



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE141599**

**OPTIMASI *MULTIOBJECTIVE UNIT COMMITMENT*  
DENGAN EMISI MEMPERTIMBANGKAN SPINNING RESERVE  
MENGUNAKAN METODE LAGRANGIAN RELAXATION**

Mochammad Rizky Pratama  
NRP 2211100107

Dosen Pembimbing  
Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.  
Dr. Rony Seto Wibowo, ST., MT.

ELECTRICAL ENGINEERING DEPARTEMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2015



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE141599**

**OPTIMASI *MULTIOBJECTIVE UNIT COMMITMENT*  
DENGAN EMISI MEMPERTIMBANGKAN SPINNING RESERVE  
MENGUNAKAN METODE LAGRANGIAN RELAXATION**

Mochammad Rizky Pratama  
NRP 2211100107

Dosen Pembimbing  
Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.  
Dr. Rony Seto Wibowo, ST., MT.

ELECTRICAL ENGINEERING DEPARTEMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2015



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE141599**

***MULTIOBJECTIVE UNIT COMMITMENT OPTIMIZATION  
WITH EMISSION BY CONSIDERING SPINNING RESERVE  
USING LAGRANGIAN RELAXATION METHOD***

Mochammad Rizky Pratama  
NRP 2211100107

Supervisor  
Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.  
Dr. Rony Seto Wibowo, ST., MT.

ELECTRICAL ENGINEERING DEPARTEMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2015

**OPTIMASI MULTIOBJECTIVE UNIT COMMITMENT DENGAN  
EMISI MEMPERTIMBANGKAN SPINNING RESERVE  
MENGUNAKAN METODE LAGRANGIAN RELAXATION**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Teknik Sistem Tenaga**

**Jurusan Teknik Elektro**

**Institut Teknologi Sepuluh Nopember**

**Menyetujui :**

**Dosen Pembimbing I**

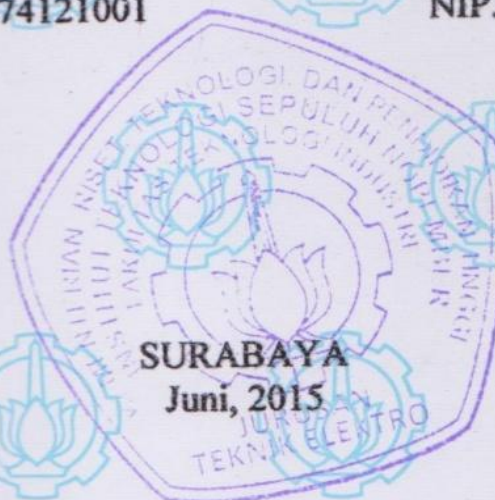


**Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D**  
**NIP. 194907151974121001**

**Dosen Pembimbing II**



**Dr. Rony Seto Wibowo, ST., MT**  
**NIP. 197411292000121001**





# Optimasi *Multiobjective Unit Commitment* Dengan Emisi Mempertimbangkan *Spinning Reserve* Menggunakan Metode *Lagrangian Relaxation*

Mochammad Rizky Pratama  
2211100107

Dosen Pembimbing I : Prof. Ir. Ontoseno P, M.Sc., Ph.D.  
Dosen Pembimbing II : Dr. Rony Seto Wibowo, ST., MT.

## Abstrak:

Pembangkit termal mempunyai peranan penting dalam hal produksi tenaga listrik. Tetapi pembangkit termal menghasilkan dampak emisi yang mempengaruhi masalah lingkungan. *Multiobjective unit commitment* merupakan salah satu solusi selain *economic dispatch*, dengan fungsi biaya untuk menekan biaya dan juga terdapat *emission dispatch* dengan fungsi emisi untuk menekan jumlah emisi pada perencanaan operasi dalam sistem tenaga. Dalam pengoperasian pembangkit juga terdapat *security constraint* berupa *spinning reserve* untuk menjaga keandalan operasi sistem. *Multiobjective function* nantinya akan dimodelkan kedalam *single objective function* dengan menggunakan *price penalty factor*. Masalah *unit commitment* akan diselesaikan dengan metode *lagrangian relaxation*. Pada tugas akhir ini penyelesaian *multiobjective unit commitment* dikerjakan pada *Visual Basic* dan akan dibuat *interface* untuk menampilkan data masukkan dan hasil akhir.

Kata Kunci : *Multiobjective Unit Commitment*, *Price Penalty Factor*, *Visual Basic*, *Spinning Reserve*, *Lagrangian Relaxation*.

# ***Optimasi Multiobjective Unit Commitment with Emission by considering Spinning Reserve using Lagrangian Relaxation Method***

M Rizky Pratama  
2211100107

Supervisor I : Prof. Ir. Ontoseno P, M.Sc., Ph.D.  
Supervisor II : Dr. Rony Seto Wibowo, ST., MT.

## **Abstract :**

*Thermal plant has a main role to bring a bright future in industrial world. And yet it has a side effect which is affecting environment. Multiobjective unit commitment is one of solutions to planning operation systems to maintain cost efficiency from cost function and emission yields from emission function. Both functions will be transformed into single objective function so that the function could be determining multiobjective unit commitment. To transform these functions, price penalty factor and weight should be included. Spinning reserve as security constraint will be involved to determine unit commitment. Unit commitment will be done with lagrangian relaxation method. And this final project uses a visual basic to create the interface and build algorithms.*

**Keywords :** *Multiobjective Unit Commitment, Price Penalty Factor, Visual Basic, Spinning Reserve, Lagrangian Relaxation.*

## KATA PENGANTAR

Pertama-tama marilah kita ucapkan puja dan puji syukur kepada Allah SWT karena limpahan rahmat yang telah diberikan oleh-Nya kepada kita semua terutama dalam penyelesaian Tugas Akhir ini dengan judul “Optimasi *Multiobjective Unit Commitment* dengan Emisi Mempertimbangkan *Spinning Reserve* Menggunakan Metode *Lagrangian Relaxation*”.

Dalam menyelesaikan Tugas Akhir ini penulis mendapatkan banyak sekali bantuan dan dukungan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang selalu meyakini penulis dalam kondisi apapun hingga dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Orang tua dan keluarga tercinta yang sangat pengertian dan mendukung penulis dalam berbagai bentuk seperti doa, moral, dan material.
3. Bapak Prof. Ir. Ontoseno Penangsang M.Sc., Ph.D., selaku dosen pembimbing 1 Tugas Akhir.
4. Bapak Dr. Rony Seto Wibowo ST., MT., selaku dosen pembimbing 2 Tugas Akhir.
5. Teman-teman seperjuangan yang saling membantu dalam penyelesaian Tugas Akhir ini.
6. Semua pihak yang tidak bisa disebutkan satu persatu yang turut membantu dalam penyelesaian Tugas Akhir ini baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa masih banyak kekurangan di dalam Tugas Akhir ini. Untuk itulah penulis menerima segala kritik dan saran yang berguna bagi penulis. Semoga buku Tugas Akhir ini berguna bagi orang lain. Sekian dan terima kasih.

Surabaya, Juni 2015

Penulis





*(Halaman ini sengaja dikosongkan)*

# TABLE OF CONTENT

<b>TITLE PAGE</b>	
<b>AUTHENTICATION</b>	
<b>APPROVAL SHEET</b>	
<b>ABSTRAK</b>	i
<b>ABSTRACT</b>	iii
<b>PREFACE</b>	v
<b>TABLE OF CONTENT</b>	vii
<b>ILLUSTRATIONS</b>	ix
<b>TABLES</b>	xi
<b>CHAPTER 1 INTRODUCTION</b>	1
1.1 Background	1
1.2 Research Aim	2
1.3 Problems	2
1.4 Problem Circumscription	2
1.5 Research Method	3
1.6 Process of Writing	4
1.7 Relevation	4
<b>CHAPTER 2 MULTIOBJECTIVE UNIT COMMITMENT WITH SPINNING RESERVE</b>	7
2.1 Generator	7
2.1.1 Generator Ability	7
2.1.2 Thermal Generator Characteristic	8
2.2 Load	9
2.2.1 Base Load	10
2.2.2 Medium Load	10
2.2.3 Peak Load	10
2.3 Unit Commitment	10
2.3.1 Minimum up-time	12
2.3.2 Minimum down-time	12
2.3.3 Spinning Reserve	12
2.3.4 Start-Up Cost	12
2.4 Unit Commitment Method	13
2.4.1 Priority List	13
2.4.2 Dynamic Programming	14
2.4.2 Lagrangian Relaxation	18
2.5 Economic Dispatch	20
2.6 Emission Dispatch	21

### **CHAPTER 3      MULTIOBJECTIVE UNIT COMMITMENT IMPLEMENTATION USING LAGRANGIAN RELAXATION METHOD .....**

3.1	Multiobjective Optimization.....	23
3.1.1	Price Penalty Factor.....	23
3.1.2	Weight .....	25
3.1.3	Multiobjective Algorithm .....	26
3.2	Lagrangian Relaxation Algorithm .....	26
3.3	Input Argument Multiobjective Unit Commitment .....	31
3.4	Syntax Software Multiobjective Unit Commitment using Delphi .....	32
3.5	Software utilizing to calculate Multiobjective Unit Commitment.....	33

### **CHAPTER 4 SIMULATION & RESULT .....37**

4.1	Case Study .....	38
4.1.1	Case 1 .....	38
4.1.2	Case 2 .....	40
4.1.3	Case 3 .....	42
4.1.4	Case 4 .....	44
4.2	Power Curve for Every Case. ....	47
4.3	Cases Analyst.....	49

### **CHAPTER 5 CONCLUSION & PROPOSITION.....53**

5.1	Conclusion .....	53
5.2	Proposition.....	53

### **REFERENCES .....55**

### **ENCLOSURE .....57**

### **BIOGRAPHY.....63**



## DAFTAR TABEL

<b>Tabel 3.1</b>	Argumen Input.....	31
<b>Tabel 3.2</b>	Sintaksis.....	32
<b>Tabel 4.1</b>	Data masukan <i>cost function</i> IEEE 39 bus.....	37
<b>Tabel 4.2</b>	Data masukan <i>emission function</i> IEEE 39 bus.....	37
<b>Tabel 4.3</b>	Data masukan minimum up- and down-time, start-up cost, dan inisial status.....	38
<b>Tabel 4.4</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BCC) Interval 1-8.....	39
<b>Tabel 4.5</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BCC) Interval 9-16.....	39
<b>Tabel 4.6</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BCC) Interval 17-24.....	40
<b>Tabel 4.7</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BEC) Interval 1-8.....	40
<b>Tabel 4.8</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BEC) Interval 9-16.....	41
<b>Tabel 4.9</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BEC) Interval 17-24.....	42
<b>Tabel 4.10</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission) Interval 1-8.....	42
<b>Tabel 4.11</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission) Interval 9-16.....	43
<b>Tabel 4.12</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission) Interval 17-24.....	44
<b>Tabel 4.13</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission with SPR) Interval 1-8.....	45
<b>Tabel 4.14</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission with SPR) Interval 9-16.....	45
<b>Tabel 4.15</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission with SPR) Interval 17-24.....	46

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Pemodelan pembangkit termal sederhana.....	7
<b>Gambar 2.2</b>	Kurva Karakteristik Input-Output Pembangkit Termal Uap.....	9
<b>Gambar 2.3</b>	Kurva Beban harian.....	9
<b>Gambar 2.4</b>	Pola sederhana <i>peak valley</i> beban.....	10
<b>Gambar 2.5</b>	Pola beban dengan aturan <i>unit commitment</i> .....	11
<b>Gambar 2.6</b>	Representasi <i>Dynamic Programming</i> .....	14
<b>Gambar 2.7</b>	Diagram alur <i>Dynamic Programming</i> .....	15
<b>Gambar 2.8</b>	Jalur pencarian DP.....	17
<b>Gambar 2.9</b>	Jalur Unit i untuk minimum <i>up- and down-time</i> .....	19
<b>Gambar 2.10</b>	Tampilan pengerjaan Delphi.....	22
<b>Gambar 3.1</b>	Diagram alur <i>price penalty factor</i> .....	24
<b>Gambar 3.2</b>	Diagram alur <i>lagrangian relaxation</i> .....	30
<b>Gambar 3.3</b>	<i>Main Display Multiobjective Unit Commitment</i> .....	33
<b>Gambar 3.4</b>	<i>Editor Input</i> .....	34
<b>Gambar 3.5</b>	<i>Input</i> untuk beban setiap interval.....	34
<b>Gambar 3.6</b>	<i>Weight Preferences Option</i> perhitungan pada <i>Software PowerGen</i> .....	34
<b>Gambar 3.7</b>	<i>Display</i> untuk <i>min up- dan down-time option</i> dan <i>method options</i> .....	35
<b>Gambar 4.1</b>	Kurva daya <i>Best Economic Commitment</i> .....	47
<b>Gambar 4.2</b>	Kurva daya <i>Best Emission Commitment</i> .....	47
<b>Gambar 4.3</b>	Kurva daya <i>Combined Economic Emission</i> .....	48
<b>Gambar 4.4</b>	Kurva daya <i>Combined Economic Emission with Spinning Reserve</i> .....	49
<b>Gambar 4.5</b>	Grafik biaya untuk kasus 1, 2 dan 3.....	49
<b>Gambar 4.6</b>	Grafik emisi untuk kasus 1, 2 dan 3.....	50

## BIOGRAFI



Penulis memiliki nama lengkap Mochammad Rizky Pratama. Penulis lahir di Samarinda tanggal 27 Juli 1994. Penulis adalah anak pertama dari pasangan Siswanto dan Tuti Alfiah. Setelah enam tahun menempuh pendidikan dasar di SDN 009 Sungai Kunjang, Penulis melanjutkan pendidikan ke SMPN 10 Samarinda pada tahun 2005. Mulai tahun 2008, penulis melanjutkan pendidikan di SMAN 3 Samarinda dan lulus pada tahun 2011. Pada tahun yang sama, penulis masuk ke Jurusan S1 Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS), Surabaya. Di semester kelima, penulis mulai mendalami Bidang Studi Sistem Tenaga. Penulis juga aktif dalam berbagai kegiatan diantaranya UKM Robotika ITS, UKM Maritime Challenge, dan Workshop Teknik Elektro ITS. Penulis dapat dihubungi melalui alamat email: [rizkypratama.elits@gmail.com](mailto:rizkypratama.elits@gmail.com)



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Energi listrik telah menjadi kebutuhan primer bagi kehidupan manusia saat ini, sehingga menyebabkan permintaan energi listrik terus meningkat setiap tahunnya. Semakin berkembangnya teknologi menjadi salah satu alasan peningkatan permintaan daya listrik ini. Selain itu terus bertambahnya jumlah penduduk turut menghadirkan masalah dalam kenaikan beban. Hal tersebut menjadi tantangan produsen untuk mengelola dan mengatur pembangkitan agar *supply* dan *demand* terpenuhi, sehingga produsen listrik tidak mengalami kerugian.

Peningkatan kebutuhan tenaga listrik harus diimbangi dengan pengelolaan sumber daya yang baik. Sehingga mampu menyediakan tenaga listrik ekonomis dan tetap menjaga kontinuitas tenaga listrik. Penjadwalan pembangkit atau *unit commitment* merupakan metode yang digunakan untuk menekan biaya yang dibutuhkan untuk membangkitkan daya listrik. Faktor biaya, pada pembangkit termal mampu mempengaruhi hasil *unit commitment*. Pada setiap persoalan *unit commitment*, setiap unit pembangkit mempunyai karakteristik yang berbeda-beda baik dari fungsi biaya, minimum *up*- dan *down-time* dan biaya *start-up* pembangkit. Metode yang digunakan pada tugas akhir ini adalah *lagrangian relaxation*. Pada metode ini selain penentuan unit mana saja yang nyala dan tidak nyala, metode ini juga mengoptimalkan daya pembangkitan setiap unit yang nyala.

Selain dari biaya operasi, faktor lain yang penting untuk dianalisis adalah faktor emisi yang berdampak pada lingkungan. Tugas akhir ini menggunakan pembangkit termal untuk membangkitkan daya listrik yang notabene menggunakan bahan bakar fosil sebagai bahan bakar utama. Pembangkit termal tersebut dapat mencemarkan lingkungan dengan emisi yang dihasilkan terutama bagi atmosfer. Maka dari itu penting untuk membangkitkan daya listrik pada pembangkit termal dengan memperhatikan tidak hanya pada segi biaya tetapi dari segi lingkungan juga. Beberapa metode sudah dilakukan oleh para peneliti untuk menyelesaikan permasalahan tersebut. Seperti melakukan teknik probabilitas untuk mendapatkan penyelesaian optimal pada satu masalah dengan dua objek. *Multiobjective* merupakan cara lain penyelesaian untuk masalah lebih dari satu fungsi objektif. Dengan adanya dua fungsi yang dioptimalkan, yaitu fungsi biaya dan fungsi emisi, kedua fungsi tersebut akan dibuat menjadi *single objective function*.

Perhitungan *multiobjective unit commitment* membutuhkan proses yang panjang oleh karena itu pada tugas akhir ini diusulkan menggunakan Delphi. Delphi merupakan bahasa pemrograman berbasis Windows yang menyediakan fasilitas pembuatan aplikasi visual. Delphi memberikan kemudahan dalam menggunakan kode program, kompilasi yang cepat, penggunaan data unit ganda untuk pemrograman modular, pengembangan perangkat lunak, pola desain yang menarik serta diperkuat dengan bahasa pemrograman yang terstruktur dalam bahasa pemrograman Object Pascal. Delphi memiliki tampilan khusus yang didukung suatu lingkup kerja komponen Delphi untuk membangun suatu aplikasi dengan menggunakan *Visual Component Library* (VCL). Sebagian besar pengembangan Delphi menuliskan dan mengkompilasi kode program dalam IDE (*Integrated Development Environment*). Sehingga pada tugas akhir ini akan menghasilkan sebuah aplikasi perhitungan yang menggunakan bahasa yang dapat dipahami oleh pengguna. Aplikasi penyelesaian *multiobjective unit commitment* dengan memperhatikan *spinning reserve* yang lebih mudah untuk berinteraksi dengan pengguna.

## 1.2 Tujuan Penelitian

Tujuan yang ingin dicapai pada Tugas Akhir ini adalah sebagai berikut:

1. Membuat aplikasi perhitungan *multiobjective unit commitment* dengan menggunakan metode *lagrangian relaxation* berbasis DELPHI.
2. Melihat pengaruh emisi dan *spinning reserve* pada perhitungan MUC

## 1.3 Permasalahan

Dalam Tugas Akhir ini diharapkan permasalahan aplikasi perhitungan *multiobjective unit commitment* yang dapat dengan mudah berinteraksi dengan pengguna dapat berjalan dengan memperhatikan *spinning reserve* yang diselesaikan dengan metode *lagrangian relaxation*.

## 1.4 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Aplikasi perhitungan dikerjakan dengan menggunakan DELPHI
2. Metode yang digunakan untuk menyelesaikan permasalahan *multiobjective unit commitment* adalah metode *lagrangian relaxation*
3. Fungsi yang diperhitungkan adalah fungsi biaya dan fungsi emisi

4. Biaya *start-up cost* merupakan *fixed cost*, tidak memperhatikan berapa lama unit tersebut telah di *shut down*.
5. Tidak memperhatikan *transmission losses*

## 1.5 Metode Penelitian

Pada tugas akhir ini akan dilakukan penelitian tentang *multiobjective unit commitment* dengan memperhatikan batasan *spinning reserve* untuk rentang waktu tertentu. *Lagrangian relaxation* digunakan untuk menentukan hasil *unit commitment*.

### 1. Studi Literatur

Literatur - literatur yang digunakan berasal dari jurnal, prosiding dan *text book*. Dalam studi literatur dipelajari teori pendukung yang berisi tentang *unit commitment* yang melihat pengaruh dari fungsi biaya, fungsi emisi dan *spinning reserve*, metode *lagrangian relaxation* yang dapat diterapkan dalam penyelesaian output pembangkitan dalam persamaan *multiobjective unit commitment* dan pembuatan *interface* menggunakan Delphi.

### 2. Pengumpulan Data dan Perhitungan

Data yang bersangkutan dalam permasalahan *unit commitment* seperti batasan maximum pembangkit, batasan minimum pembangkit, koefisien fungsi bahan bakar, koefisien fungsi emisi, data beban tiap periode dalam interval waktu 1 jam, *fuelcost*, minimum *up-time*, minimum *down-time*, dan *spinning reserve* serta bobot. Berdasarkan formulasi yang telah diketahui dilakukan perhitungan terlebih dahulu untuk menunjang pemodelan dan simulasi.

### 3. Perencanaan dalam Pembuatan Program

Dari data perhitungan tersebut dibuatlah strukur perancangan logika berupa diagram alir untuk proses pembuatan program yang nantinya akan diimplementasikan kedalam bahasa pemrograman Delphi.

### 4. Simulasi Analisa Data

Setelah pembuatan program telah selesai dan menjadi sebuah aplikasi perhitungan, simulasi akan dicoba dengan menggunakan data inputan yang telah tersedia. Simulasi bertujuan untuk melihat aplikasi perhitungan *multiobjective unit commitment* yang dibuat telah sesuai dengan teori perhitungan yang ada. Dari penjalanan simulasi akan didapatkan data yang akan dianalisa kebenarannya. Analisa bertujuan untuk memastikan hasil perhitungan program telah sesuai dengan metode yang digunakan. Untuk mendapatkan hasil perhitungan yang sesuai, hasil



dari data outputan akan dibandingkan dengan data *output* referensi dan batasan yang telah ditentukan.

## 5. Penulisan Buku

Hasil penelitian yang telah dilakukan akan dilaporkan dalam sebuah buku laporan Tugas Akhir. Isi dari laporan berdasarkan kesimpulan dari analisis yang telah didapat beserta tahapan yang ada di dalamnya.

### 1.6 Sistematika Penulisan

#### BAB 1 PENDAHULUAN

Bab ini membahas Latar Belakang, Tujuan Penelitian, Permasalahan, Batasan Masalah, Metode Penelitian, Sistematika Penulisan, dan Relevansi pada laporan Tugas Akhir.

#### BAB 2 MULTIOBJECTIVE UNIT COMMITMENT

Bab ini berisikan teori penunjang dalam Tugas Akhir. Teori yang dibahas meliputi Generator, Beban, Karakteristik Pembangkit, *unit commitment*, *spinning reserve*, *dynamic programming*, *lagrangian relaxation*, *economic dispatch*, *emission dispatch*, dan bahasa pemrograman VB/Delphi

#### BAB 3 IMPLEMENTASI MULTIOBJECTIVE UNIT COMMITMENT DENGAN METODE LAGRANGIAN RELAXATION

Bab ini berisikan penjelasan mengenai tentang *multiobjective optimization* dengan *price penalty factor* serta penggunaan *dynamic programming* dan *lagrangian relaxation* yang diterapkan pada permasalahan *multiobjective unit commitment*.

#### BAB 4 SIMULASI DAN ANALISIS

Bab ini berisi data hasil *running* aplikasi MUC yang hasil keluaran data dianalisis dengan cara membandingkan hasil *output* setiap kasus.


#### BAB 5 PENUTUP

Bab terakhir ini memberikan kesimpulan dari simulasi dan analisis yang telah dilakukan dalam Tugas Akhir.

### 1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat memberikan kontribusi sebagai berikut:

1. Dapat memberikan manfaat dalam kemudahan melakukan perhitungan optimisasi *multiobjective unit commitment* menggunakan *lagrangian relaxation* dengan memperhatikan *spinning reserve*.
2. Dapat menambah penguasaan ilmu dan teknologi di bidang optimisasi pembangkitan sistem tenaga listrik.

- 
3. Dapat menjadi referensi Tugas Akhir untuk mahasiswa yang akan mengambil Tugas Akhir untuk mengembangkan aplikasi perhitungan *multiobjective unit commitment* dengan permasalahan yang berbeda.



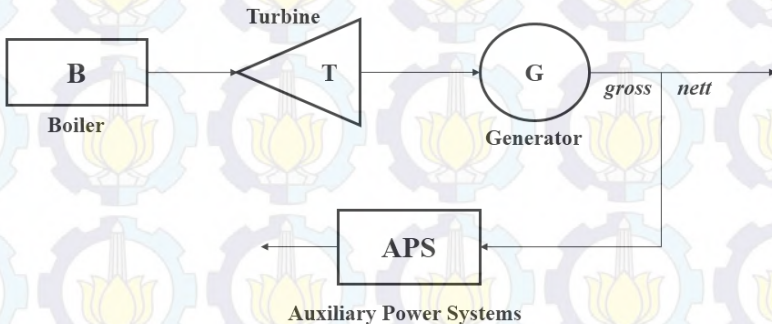


## BAB 2

# MULTIOBJECTIVE UNIT COMMITMENT DENGAN SPINNING RESERVE

### 2.1 Generator

Di sistem tenaga listrik generator merupakan salah satu komponen penting untuk mengoperasikan sistem tenaga listrik. Generator merupakan pembangkit listrik yang menghasilkan output daya untuk disalurkan ke beban. Generator yang paling umum digunakan ialah generator 3 fasa atau generator sinkron. Dalam sebuah unit pembangkitan, beberapa generator dioperasikan secara paralel dan dihubungkan ke bus jaringan untuk memenuhi *demand load*. Dari semua elemen dalam sistem tenaga listrik, biaya bahan bakar pada generator merupakan elemen terbesar dari semua jenis biaya yang diperlukan dalam sebuah usaha pembangkitan tenaga listrik. Sehingga pengoptimalan *scheduling* di semua unit pembangkit pada sistem tenaga listrik sangat diperlukan untuk menekan biaya bahan bakar agar mendapatkan nilai dari fungsi biaya pembangkitan yang minimum.



**Gambar 2.1** Pemodelan pembangkit termal sederhana

#### 2.1.1 Batasan kemampuan generator

Kemampuan generator untuk membangkitkan daya listrik mempunyai aturan yang harus diperhatikan.  $P_{min}$  dan  $P_{max}$  pada generator merupakan aturan yang harus dipenuhi untuk syarat membangkitkan daya listrik. Generator harus membangkitkan daya dalam rentang  $P_{min}$  dan  $P_{max}$  agar pengoperasian generator lebih ekonomis, efisien dan tidak menimbulkan kerusakan pada generator. Generator tidak boleh dioperasikan dibawah  $P_{min}$  (*under operation*) dan diatas  $P_{max}$  (*over operation*) seperti pada persamaan(2.0).

$$P_{i_{min}} \leq P_{gen_i} \leq P_{max_i} \quad (2.0)$$

Keterangan :

$P_{i_{min}}$  : Daya minimum unit i

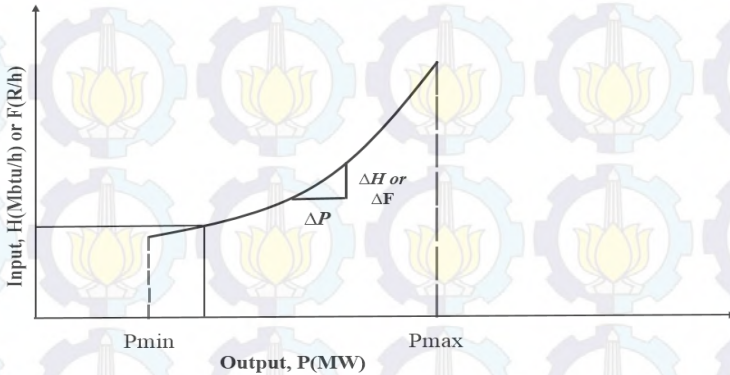
$P_{max_i}$  : Daya maksimum unit i

$P_{gen_i}$  : Daya yang dibangkitkan unit i

### 2.1.2 Karakteristik Pembangkit Termal

Hal dasar yang perlu diperhatikan untuk mendapatkan solusi ekonomis dari pembangkit termal adalah menghitung karakteristik *input-output* pada pembangkit termal. Untuk menghitung biaya pengeluaran pada pembangkit yang beroperasi, ada cukup banyak parameter yang perlu diperhatikan. Karakteristik input-output suatu pembangkit tidak hanya mencakup pada sistem tenaga listrik, tetapi terdapat juga elemen pendukung yang mempengaruhi karakteristik input-output. Biaya *maintenance* dan biaya operasi sudah termasuk dalam karakteristik input-output biaya operasi, *R per hour*, setiap unit. Selain itu juga terdapat biaya tambahan yang dimasukkan ke dalam *fixed cost*.

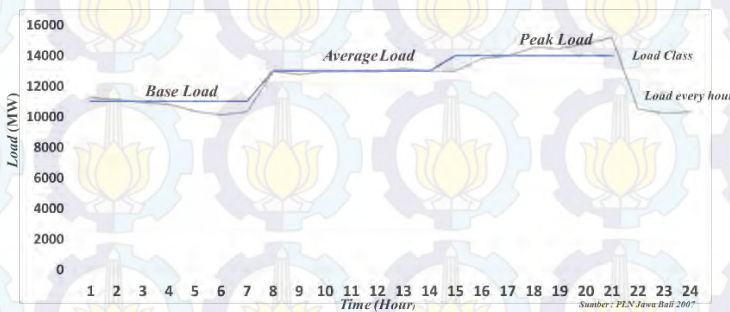
Gambar 2.2 menunjukkan karakteristik input-output dalam bentuk ideal berupa kurva *smooth convex*. Pada elemen input unit di Gambar 2.2 ditunjukkan bahwa input dapat dijadikan sebagai jumlah energi panas yang dibutuhkan per jam atau sebagai total biaya per jam. Dan pada output berupa besar daya yang dihasilkan tiap unit. Data karakteristik input-output didapat dari perhitungan-perhitungan sistem pembangkit atau dari uji tingkat panas. Pada Gambar 2.3, kurva menunjukkan hubungan perubahan daya pembangkitan yang dihasilkan dengan konsumsi bahan bakar yang dibutuhkan. *Incremental rate* =  $\Delta H / \Delta P$  atau *incremental rate* =  $\Delta F / \Delta P$ . *Incremental rate* dinyatakan dengan satuan Btu/kWh( $\Delta H / \Delta P$ ) atau R/kWh( $\Delta F / \Delta P$ ).



**Gambar 2.2** Kurva Karakteristik Input-Output Pembangkit Termal Uap

## 2.2 Beban

Salah satu yang menjadi tolak ukur kualitas sistem tenaga listrik dikatakan baik apabila pada sisi beban dikatakan terlayani dengan baik. Beban merupakan tujuan akhir dari daya listrik yang dibangkitkan oleh generator. Menurut pola konsumsi beban umumnya dibagi menjadi beberapa jenis yaitu, beban industri, beban perumahan dan beban komersil. Pada beban komersil dan beban perumahan biasanya terdiri dari sistem penerangan dan pendinginan dalam area yang luas. Beban ini tidak bergantung pada frekuensi karena mengonsumsi daya reaktif yang sangat kecil. Pola pembebanan pembangkit umumnya didasarkan pada fleksibilitas operasi, level biaya dan kapasitas pembangkitan karakteristik pembangkit. Menurut level, beban dikategorikan sebanyak tiga kategori antara lain:



**Gambar 2.3** Kurva Beban harian



### 2.2.1 Beban Dasar

Pada beban dasar, pembangkit yang dinyalakan pada sistem ialah merupakan kategori pembangkit yang memiliki daya keluaran besar, biaya kapital tinggi dan biaya operasi rendah. Pembangkit *base load* adalah pembangkit dengan 5000 jam operasi rata-rata per tahun dengan *capacity factor* lebih dari 57%.

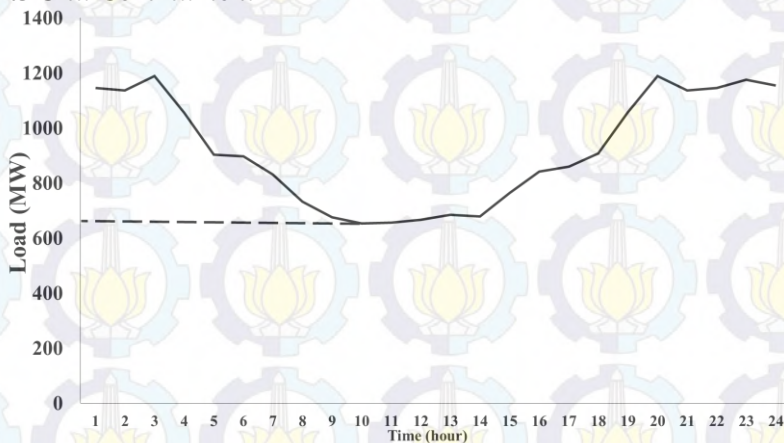
### 2.2.2 Beban Menengah

Pada level beban menengah, pembangkit yang beroperasi umumnya merupakan pembangkit *combined cycle*, pembangkit berbahan bakar gas dan pembangkit tua yang efisiensinya sudah menurun. Pembangkit *mid-range load* rata-rata beroperasi selama 2000 jam per tahun dengan  $23\% < \text{capacity factor} < 57\%$ .

### 2.2.3 Beban Puncak

Beban puncak ialah nilai beban tertinggi pada sistem jaringan tenaga listrik. Pada beban puncak pembangkit *peak load* dioperasikan untuk memenuhi beban pada waktu beban maksimum (beban puncak). Periode beban puncak tidak selalu sama. Pembangkit yang beroperasi pada beban puncak merupakan pembangkit dengan karakteristik biaya operasi yang tinggi dan biaya kapital rendah. Pengoperasian pembangkit *peak load* rata-rata beroperasi selama kurang dari 2000 jam. Pembangkit *peak load* menggunakan pembangkit listrik berbahan minyak, air dan *pump storage*.

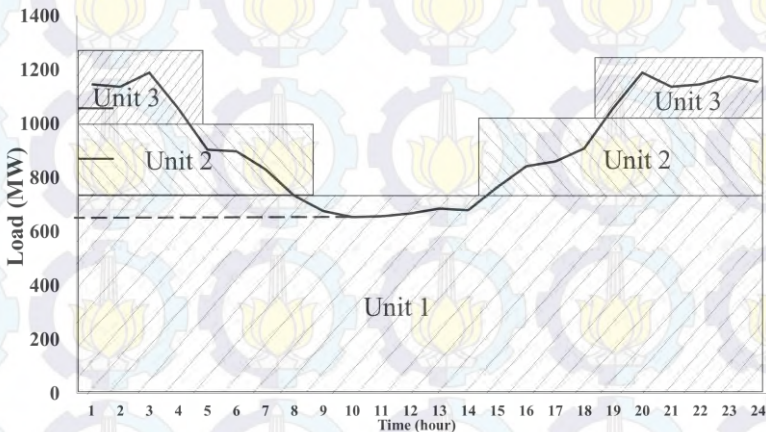
## 2.3 Unit Commitment



Gambar 2.4 Pola sederhana *Peak valley* beban

Siklus kerja manusia untuk melakukan kegiatan berdampak langsung kepada penggunaan energi listrik. Sehingga hal ini menimbulkan siklus harian, bulanan bahkan tahunan. Kasus umum yang terjadi pada siklus harian sistem tenaga listrik, pada siang hari hingga menjelang malam umumnya jumlah permintaan listrik bernilai rendah bila dibandingkan dengan konsumsi daya listrik pada malam hari, dikarenakan siklus industri yang bekerja pada waktu siang hari. Dan pada akhir minggu jumlah penggunaan listrik cenderung berkurang dibandingkan dengan hari-hari kerja.

Akibat dari siklus penggunaan listrik maka diperlukan adanya *unit commitment*. *Unit commitment* merupakan susunan penjadwalan setiap unit pembangkit yang terhubung satu sama lain untuk setiap periode dikarenakan permintaan beban yang berubah-ubah. *Unit commitment* merupakan formulasi dari kombinasi dua sub-masalah yaitu menentukan unit-unit yang aktif dan tidak aktif untuk setiap interval dan jumlah daya yang dibangkitkan setiap unit yang aktif. Dengan adanya *unit commitment* jumlah *cost* yang dapat dihemat cukup besar. Gambar 2.5 menunjukkan penjadwalan unit yang sederhana dengan melihat *demand load*. ditunjukkan bahwa unit 1 merupakan penopang untuk beban dasar. Unit 2 dinyalakan ketika unit 1 sudah tidak mampu menopang *average load* sehingga unit 2 dinyalakan dan untuk beban puncak, semua unit dinyalakan untuk membangkitkan daya listrik yang dibutuhkan untuk memenuhi *demand load*.



**Gambar 2.5** Pola beban dengan aturan *unit commitment*



Tetapi *unit commitment* mempunyai *constraint* yang harus diperhatikan antara lain adalah :

### 2.3.1 Minimum up-time

Setelah unit beroperasi, unit tersebut tidak dapat dimatikan secepatnya, hal ini dikarenakan perubahan temperatur yang perlahan sehingga menghalangi *decommit* unit pada beberapa waktu interval.

### 2.3.2 Minimum down-time

Unit tidak dapat dinyalakan seketika setelah unit *decommit*, adanya minimum waktu membuat unit bisa di nyalakan lagi setelah memenuhi interval waktu tersebut.

### 2.3.3 Spinning Reserve

Keandalan sebuah sistem tenaga listrik diukur dari jumlah beban yang diminta dan jumlah daya yang dikirimkan, apabila selisih *demand load* dan daya yang ditransmisikan berjumlah mendekati nol maka sistem dalam keadaan yang baik. Tetapi apabila pada sistem terdapat gangguan pada satu unit yang mengakibatkan unit tersebut tidak bekerja, maka dapat mengakibatkan *drop* frekuensi yang sangat besar. *Spinning reserve* dideskripsikan sebagai jumlah semua pembangkitan unit yang mungkin dapat dilakukan dikurangi dengan beban saat ini dan *transmission loss*. Pada sistem, *spinning reserve* harus ditempatkan pada lokasi yang menuntut respon yang cepat ketika terdapat satu unit yang hilang.

$$P_{i\min}^t \leq P_{i\text{generate}}^t + SPR_i \leq P_{i\max}^t \quad (2.2)$$

Keterangan :

- $P_{i\min}^t$  : Daya minimum yang dapat dibangkitkan unit i saat t  
 $P_{i\max}^t$  : Daya maksimum yang dapat dibangkitkan unit i saat t  
 $P_{i\text{generate}}^t$  : Daya yang dibangkitkan unit i saat t  
 $SPR_i$  : *Spinning reserve* unit i

### 2.3.4. Start-Up Cost

Biaya *start-up* pada pembangkit merupakan besarnya biaya yang diperlukan untuk menyalakan unit pembangkit saat interval sebelumnya sedang *decommit* dan saat interval selanjutnya unit harus *commit*. Jumlah *start-up cost* bisa bervariasi dari biaya maksimum *cold start* hingga biaya yang lebih kecil jika unit sudah *decommit* dengan interval yang kecil. Ada dua tindakan pada unit termal saat *decommit*. Pertama unit dibiarkan

dingin untuk beberapa interval dan dipanaskan kembali untuk beroperasi saat *commit*. Kedua unit diberikan energi yang cukup untuk menjaga temperatur agar biaya *start-up* unit murah.

## 2.4 Metode *Unit Commitment* [1]

Dalam penyelesaian *unit commitment* terdapat banyak metode yang bisa digunakan untuk mendapatkan hasil *unit commitment*, *economic dispatch*, *emission dispatch* dan *combined economic emission dispatch*.

### 2.4.1 *Priority List*

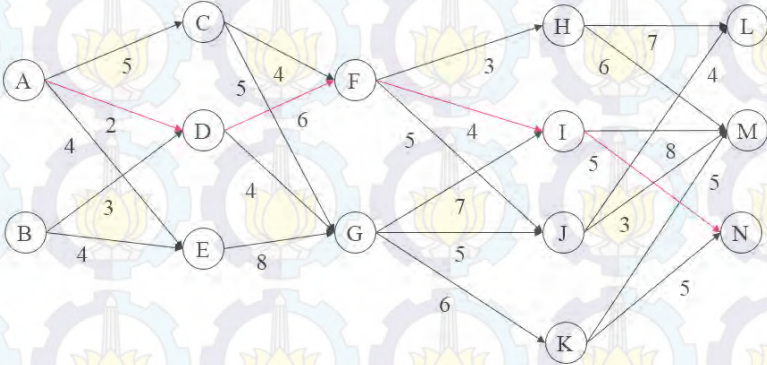
Metode solusi penyelesaian *unit commitment* yang paling sederhana adalah pembuatan *priority list* untuk unit-unit pembangkit pada sistem. Skema *priority list* dibuat dengan algoritma yang sangat sederhana seperti berikut :

- Pada setiap periode ketika beban turun, harus ditentukan untuk memilih unit, A, yang harus di *shut down*, dengan memperhatikan *spinning reserve*, pada *priority list* agar nilai *supply* dan *demand* seimbang. Jika tidak maka operasi berjalan seperti sebelumnya.
- Tentukan jumlah periode, H, sebelum unit A yang mati dinyalakan kembali.
- Jika H bernilai kurang dari minimum *down time* unit A saat unit A dibutuhkan kembali maka unit tetap dalam keadaan “on” saat terakhir kali unit A dimatikan.
- Hitung kedua biaya yang didapat, pertama jumlah biaya produksi untuk periode H selanjutnya dengan unit A yang menyala. Lalu hitung kembali jumlah biaya produksi untuk periode H selanjutnya dengan unit A yang dimatikan dengan tambahan *start-up cost*. Bandingkan kedua biaya produksi tersebut dan pilih yang paling murah.

Skema *priority list* dapat ditingkatkan dengan berbagai cara salah satunya membuat group-group yang berisilebih dari satu unit dengan berbagai *constraint* yang ada.



### 1.4.2 Dynamic Programming



**Gambar 2.6** Representasi *Dynamic Programming*

Pada *dynamic programming* untuk *unit commitment*, pendekatan yang dilakukan adalah :

1. *State* terdiri dari susunan beberapa unit dengan spesifikasi unit yang beroperasi dan tidak beroperasi.
2. Biaya *start-up* tidak bergantung pada berapa lama waktu unit tersebut telah di *shut-down*.
3. Tidak ada biaya untuk melakukan *unit decommitment*.
4. Setiap interval *state* harus memenuhi spesifikasi minimum kapasitas jumlah untuk unit yang dioperasikan.

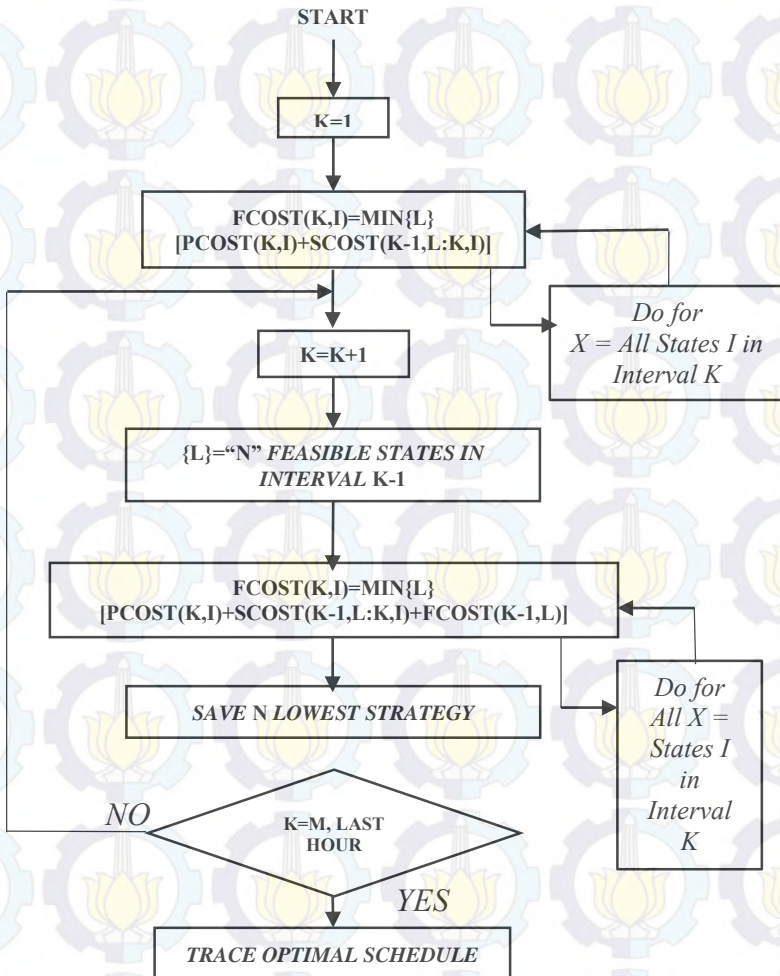
Sebuah solusi dari *dynamic programming* adalah kumpulan dari *state* pada interval pertama hingga interval terakhir. Dan *state* yang menjadi solusi dari permasalahan merupakan *feasible state*. Dan jumlah maksimum *state* yang dapat dimasukkan ke dalam penyelesaian *dynamic programming* secara matematis adalah sebagai berikut:

$$\text{Maks.State} = 2^n - 1 \quad (2.3)$$

Dimana  $n$  merupakan jumlah unit yang terhubung ke sistem dan bisa beroperasi.

*Dynamic programming* yang digunakan pada *unit commitment* adalah *forward dynamic programming*. Pada *forward dynamic programming* pendekatan yang dilakukan lebih menguntungkan apabila dibandingkan dengan *backward dynamic programming*. Pendekatan pada *forward dynamic programming* dilakukan dengan pengecekan hasil pada

state sebelumnya dimulai dari waktu inisial hingga akhir interval. Alasan praktis lain untuk menggunakan *forward dynamic programming* adalah mudah untuk menspesifikasikan constraint dan mengkomputasi perhitungan dimulai pada kondisi inisial.



**Gambar 2.7** Diagram alur *dynamic programming* [1]

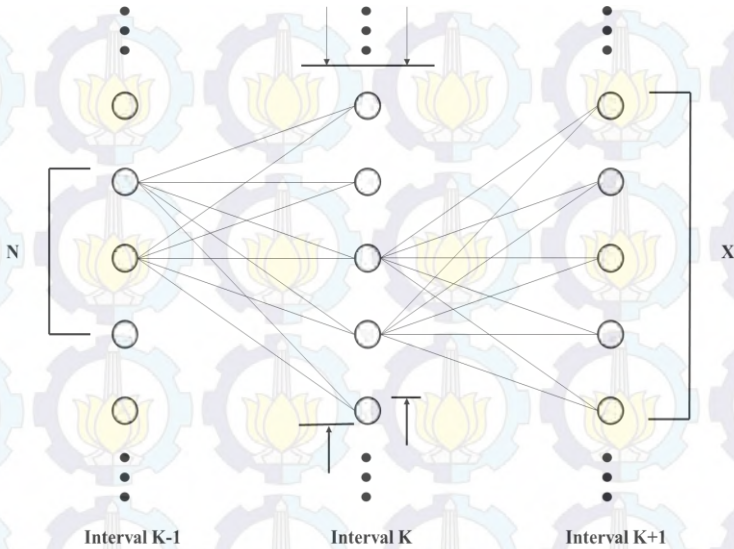


Pada Gambar 2.7 state (K,I) merupakan kombinasi unit pembangkit ke-I saat interval K. Untuk pendekatan *forward dynamic programming* kita mendefinisikan strategi sebagai transisi atau jalur, dari state-state sebelumnya ke state interval selanjutnya. Ada dua variabel pada *forward dynamic programming*, yang pertama adalah X. X merupakan jumlah kombinasi unit yang dicari pada interval tersebut. Yang kedua adalah N, merupakan strategi atau jumlah kombinasi unit yang disimpan pada setiap step untuk menentukan apakah state sebelumnya merupakan penyelesaian yang optimal untuk interval selanjutnya.

Jumlah nilai X dan N maksimal yang dapat dilakukan dalam penyelesaian juga sebesar jumlah maksimal state pada permasalahan. Akan tetapi jumlah kombinasi yang dapat dilakukan setiap interval harus merupakan state yang *feasible* untuk konstrain yang berlaku. Gambar 3.3 merupakan contoh sederhana untuk menggunakan *forward dynamic programming*. Pada Gambar 2.8, jumlah state, X, yang digunakan untuk solusi optimal berjumlah 5 dan jumlah kombinasi unit yang disimpan, N, untuk mendapatkan hasil atau jalur yang optimal beserta konstrain yang terpenuhi berjumlah 3.

*Dynamic programming* tidak hanya berperan untuk menentukan jumlah state yang akan dioptimalkan tetapi juga mengatasi *constraint* yang ada pada *unit commitment* seperti minimum *up-* dan *down-time*. Setelah semua state yang *feasible* sudah dioptimalkan maka *dynamic programming* akan mencari hasil fungsi yang paling optimal dengan cara menentukan jalur strategi mulai dari inisial interval hingga akhir interval.

Pada *unit commitment*, pendekatan *dynamic programming* yang dilakukan ialah *forward DP*. Hal ini dikarenakan pada *unit commitment* terdapat *start-up cost* yang menambah total biaya apabila terdapat unit yang harus dinyalakan.



**Gambar 2.8** Jalur pencarian DP [1]

Alasan lain *forward* DP digunakan ialah pada *unit commitment* terdapat inisial status per unit yang mengharuskan perhitungan dimulai dari awal agar memudahkan perhitungan.

$$F_{cost}(K, I) = \min[P_{cost}(K, I) + S_{cost}(K - 1, L: K, 1) + F_{cost}(K - 1, L)] \quad (2.4)$$

Keterangan:

- |                            |  |
|----------------------------|--|
| K                          | : Periode atau jam ke K  |
| I                          | : Kombinasi semua unit <i>state</i> I                                      |
| $F_{cost}(K, I)$           | : Total biaya optimum yang diperlukan untuk sampai ke state I pada stage K |
| $P_{cost}(K, I)$           | : Biaya produksi pada state I untuk stage K                                |
| $S_{cost}(K - 1, L: K, I)$ | : Transisi biaya dari state $(K - 1)$ , stage L ke state K, stage I        |

Gambar 2.4 menunjukkan cara kerja *forward dynamic programming*. Pada Gambar 2.4, A,B,C,D,...,N merupakan state untuk setiap stage dan panah merupakan harga yang dilalui dari state periode sebelum ke state periode setelahnya. Dan panah yang berwarna merah merupakan solusi untuk harga termurah mulai dari inisial status hingga *final* status untuk penyelesaian *forward dynamic programming*.

### 1.4.3 Lagrangian Relaxation

Pada metode *lagrangian relaxation* nilai inisial lambda akan ditentukan terlebih dahulu. Tentunya nilai dari lambda pertama bukanlah hasil yang benar. Ketika nilai total dari  $P_1 + P_2 + P_3 + \dots P_i < P_{target}$  maka nilai  $\lambda$  untuk iterasi berikutnya akan bertambah lebih besar dari nilai  $\lambda$  sebelumnya. Dan sebaliknya, jika nilai total  $P_1 + P_2 + P_3 + \dots P_i > P_{target}$  maka nilai  $\lambda$  untuk iterasi berikutnya akan lebih kecil daripada nilai dari  $\lambda$  sebelumnya. Proses ini akan melakukan itersi nilai  $\lambda$  hingga mendapatkan hasil dimana  $P_1 + P_2 + P_3 + \dots P_i = P_{target}$ .

Dasar dari metode *lagrangian relaxation* adalah pendekatan dual optimisasi dari *lagrange multiplier*. *Lagrange multiplier* menggunakan teknik yang menyelesaikan variabel lagrange, berupa lambda, dan setelahnya menemukan solusi dari variabel permasalahan yang ingin ditemukan.

*Objective function :*

$$\sum_{t=1}^T \sum_{i=1}^N [F_i(P_i^t) + Start\ up\ cost_{i,t}] U_i^t = F(P_i^t, U_i^t) \quad (2.5)$$

Keterangan :

$U_i^t$  : “on” atau “off” unit i saat t  
 $P_i^t$  : Daya yang dibangkitkan unit i saat t  
 $Start\ up\ cost_{i,t}$  : Biaya untuk menyalakan unit i saat t

Pada solusi *lagrangian relaxation* ada tiga hal lain yang harus diperhatikan setelah fungsi objektif yaitu :

1. Konstrain Beban

$$P_{load}^t - \sum_{i=1}^N P_i^t U_i^t = 0 \text{ for } t = 1 \dots T \quad (2.6)$$

2. Batas Unit

$$U_i^t P_i^{min} \leq P_i^t \leq U_i^t P_i^{max} \text{ for } i = 1 \dots N \text{ and } t = 1 \dots T \quad (2.7)$$

3. Unit minimum up- and down-time



Fungsi objektif pada persamaan (2.5) dapat di ubah menjadi persamaan lagrange dengan memperhatikan *constraint* yang ada untuk menemukan solusi dari *economic dispatch*.

$$\mathcal{L}(P, U, \lambda) = F_i^t(P_i^t) + \sum_{t=1}^T \lambda^t (P_{load}^t - \sum_{i=1}^N P_i^t U_i^t) \quad (2.8)$$

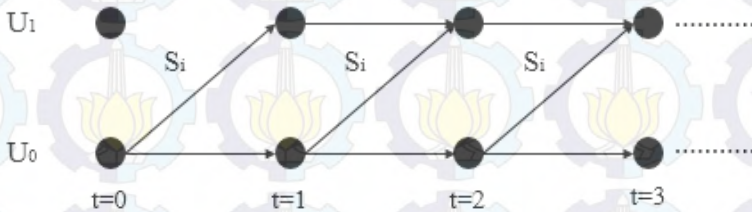
Dengan metode *lagrangian relaxation* kita mendapatkan solusi penyelesaian dari setiap unit secara terpisah, tidak memperhatikan unit lain. Persoalan *unit commitment* mengharuskan kita untuk meminimisasi fungsi lagrange persamaan (2.8). Prosedur *lagrangian relaxation* dilakukan melalui dengan prosedur *dual optimization*. Prosedur *dual optimization* dipaksa untuk mendapatkan hasil optimum dengan cara memaksimalkan fungsi lagrange dengan memperhatikan *lagrangian multipliers*, saat ingin meminimisasi variabel lain.

$$q(\lambda) = \min \mathcal{L}(P, U, \lambda) \quad (2.9)$$

$$q^*(\lambda) = \max q(\lambda) \quad (2.10)$$

Dengan persamaan (2.8), fungsi objektif setiap unit diselesaikan secara terpisah, tanpa memperhatikan fungsi objektif lain.

Ada dua *constraint* yang dimasukkan ke dalam persamaan (2.5) yaitu persamaan (2.7) dan minimum *up-* dan *down-time*. Untuk *constraint* minimum *up-* dan *down-time* dapat diselesaikan dengan *dynamic programming*, kemungkinan unit  $U_i^t$  hanya bernilai 1 dan 0, saat  $U_i^t=0$  maka unit tidak dioptimisasi dan saat  $U_i^t=1$  maka fungsi objektif unit i di optimisasi, seperti ditunjukkan pada Gambar 2.7.



**Gambar 2.9** Jalur Unit i untuk minimum *up- and down-time*

## 2.5 Economic Dispatch

*Economic dispatch* merupakan salah satu aspek yang perlu diperhatikan untuk mendapatkan biaya pembangkitan yang optimal. *Economic dispatch* ialah perhitungan output daya setiap pembangkit untuk beroperasi secara ekonomis pada kondisi beban setiap periode. ED merupakan optimasi *non-linear* yang variabel outputnya adalah keluaran daya pada setiap unit yang aktif. Setiap unit pembangkit mempunyai karakteristik fungsi biaya terhadap daya yang berbeda-beda, yang dipengaruhi oleh efisiensi unit pembangkit dan jenis bahan bakar yang digunakan. Secara matematis fungsi biaya adalah sebagai berikut :

$$\sum_{i=1}^n H_i = a_i P_i^2 + b_i P_i + c_i \quad (2.11)$$

$$\sum_{i=1}^n F_i = \sum_{i=1}^n f_i \times \sum_{i=1}^n H_i \quad (2.12)$$

$$\begin{aligned} F_T &= F_1 + F_2 + F_3 + \dots + F_n \\ &= \sum_{i=1}^n F_i \end{aligned} \quad (2.13)$$

Keterangan :

$H_i$	: Input kalor per unit (Btu/h or Mbtu/h)
$P_i$	: Daya operasi setiap unit pembangkit
$a_i$	: Koefesien orde-2 setiap unit
$b_i$	: Koefesien orde-1 setiap unit
$c_i$	: Koefesien <i>No-Load Cost</i> setiap unit
$F_i$	: Biaya yang diperlukan untuk membangkitkan $P_i$
$f_i$	: <i>Fuel Cost</i> (R per Btu / R per Mbtu)
$F_T$	: Total keseluruhan biaya semua unit

Secara matematis persamaan *economic dispatch* (2.12) dapat dipersingkat ke persamaan (2.13). Sistem yang di *dispatch* terdiri dari N unit pembangkit termal yang terhubung ke *single bus-bar* yang menyediakan daya untuk  $P_{load}$ .

## 2.6 Emission Dispatch

Saat beroperasi pembangkit listrik thermal, berbahan bakar fosil, akan menghasilkan emisi sebagai *side effect*. Emisi yang dihasilkan setiap unit pembangkit dapat dimodelkan secara matematis, sebagai berikut :

$$\sum_{i=1}^n E_i = \alpha_i P_i^2 + \beta_i P_i + \gamma_i \quad (2.14)$$

Keterangan :

$E_i$  : Emisi (kg/h)

$P_i$  : Daya operasi setiap unit pembangkit

$\alpha_i$  : Koefesien emisi

$\beta_i$  : Koefesien emisi

$\gamma_i$  : Koefesien emisi





*(Halaman ini sengaja dikosongkan)*

## BAB 3

### IMPLEMENTASI *MULTIOBJECTIVE UNIT COMMITMENT* DENGAN METODE *LAGRANGIAN RELAXATION*

#### 3.1 *Multiobjective Optimization*

*Multiobjective Optimization* merupakan penyelesaian dari beberapa fungsi tujuan. Untuk menyelesaikan fungsi-fungsi tujuan yang ingin dicari terdapat banyak cara, salah satunya dengan mengkonversikan fungsi-fungsi tersebut menjadi satu, lalu dengan fungsi baru tersebut dicari nilai optimal pada fungsi baru. Dan setelah didapatkan *variabel* yang optimal pada fungsi baru, *variabel* yang didapatkan, dimasukkan ke dalam masing-masing fungsi. Secara umum bentuk matematis untuk *multiobjective optimization* adalah :

$$f(f_1(x), f_2(x), \dots, f_i(x)) \quad (3.1)$$

Dimana  $i$  merupakan jumlah fungsi yang ingin di gabung.

##### 3.1.1 *Price Penalty Factor* [6]

Pada tugas akhir ini fungsi baru yang ingin dibuat terdiri dari fungsi biaya dan fungsi emisi. *Economic dispatch* dan *emission dispatch* merupakan dua masalah yang berbeda. Jadi fungsi *economic* dan fungsi *emission* akan dikonversikan kedalam *single objective function* dengan menggunakan *price penalty factor*.

$$TC = FC(P_g) + h \times EC(P_g) \quad (3.2)$$

Dimana  $TC$  merupakan total biaya operasional pada sistem dengan *constraint* yang ada,  $FC$  merupakan fungsi biaya dan  $EC$  merupakan fungsi emisi. Dan  $h$  merupakan *price penalty factor*. *Price penalty factor* dapat di cari dengan langkah sebagai berikut :

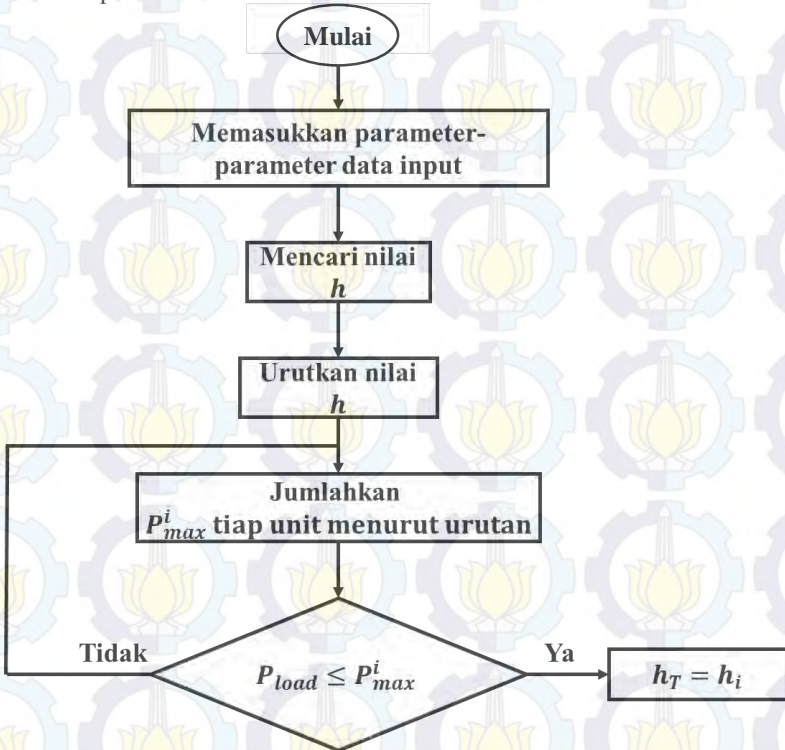
1. Hitung biaya pembangkitan unit pembangkit pada daya maksimal.
2. Hitung jumlah emisi unit pembangkit pada daya maksimal.
3. Bagi biaya pembangkitan dengan jumlah emisi pada setiap unit pembangkit.

$$h_i = \frac{FC_i(P_{\max})}{EC_i(P_{\max})}, h_i = 1, 2, 3, \dots, n \text{ R/kg} \quad (3.3)$$

4. Urut nilai *price penalty factor* mulai dari yang terkecil hingga terbesar.
5. Jumlahkan semua pembangkitan maksimal pada setiap unit mengikuti urutannya hingga

$$P_{load}^t \leq P_{\max}^i \quad (3.4)$$

6. Setelah memenuhi pada persamaan (3.4) maka nilai  $h_i$  yang akan digunakan untuk  $P_{load}^t$  adalah perhitungan  $h_i$  dengan penambahan unit terakhir.



Gambar 3.1 Diagram alur price penalty factor



### 3.1.2 Bobot

Pada masalah *multiobjective optimization*, bobot merupakan indikasi akan dibawa kemana penyelesaian multiobjektif tersebut. Pada *multiobjective unit commitment* bobot diletakkan kedalam persamaan emisi dan biaya dengan total semua bobot pada masing-masing persamaan bernilai 1. Secara matematis bobot dapat di rumuskan sebagai berikut :

$$\varphi_c(P_i) = w_{fuel}F_c(P_i) + w_{emi}E_c(P_i) \quad (3.5)$$

Untuk perhitungan data pada bobot terdapat tiga kasus yaitu

1. *Economic Dispatch*

Pada *Economic Dispatch*  $w_{fuel} = 1$  dan  $w_{emi} = 0$

$$\begin{aligned} \varphi_c(P_i) &= (1)F_c(P_i) + (0)E_c(P_i) \\ \varphi_c(P_i) &= F_c(P_i) \end{aligned} \quad (3.6)$$

Dari persamaan (3.6) maka fungsi yang akan diperhatikan adalah fungsi ekonomi saja, sehingga hasil yang akan didapatkan lebih cenderung untuk optimal pada sisi biaya saja, sedangkan pada sisi emisi lebih cenderung belum optimal.

2. *Emission Dispatch*

Pada *Emission Dispatch*  $w_{fuel} = 0$  dan  $w_{emi} = 1$

$$\begin{aligned} \varphi_c(P_i) &= (0)F_c(P_i) + (1)E_c(P_i) \\ \varphi_c(P_i) &= E_c(P_i) \end{aligned} \quad (3.7)$$

Dari persamaan (3.7) maka fungsi yang akan diperhatikan adalah fungsi emisi saja, sehingga hasil yang akan didapatkan lebih cenderung untuk optimal pada sisi emisi saja, sedangkan pada sisi ekonomi lebih cenderung belum optimal.

3. *Combined Economic Emission Dispatch*

Untuk kombinasi *economic* dan *emission dispatch* maka  $w_{fuel} = 1$  dan  $w_{emi} = 1$

### 3.1.3 Algoritma Multiobjective

Alur perhitungan *multiobjective* pada aplikasi *software* dimulai dengan mengambil semua data input untuk dilakukan perhitungan untuk menjadikan dua *objective function* menjadi *single objective function*.

$$\begin{bmatrix} \text{coeff } a_i \\ \text{coeff } b_i \\ \text{coeff } c_i \end{bmatrix} = \begin{bmatrix} \text{coeff } a_i^{\text{economic}} \\ \text{coeff } b_i^{\text{economic}} \\ \text{coeff } c_i^{\text{economic}} \end{bmatrix} + h_i \begin{bmatrix} \text{coeff } a_i^{\text{emission}} \\ \text{coeff } b_i^{\text{emission}} \\ \text{coeff } c_i^{\text{emission}} \end{bmatrix} \quad (3.8)$$

Lalu dimasukkan ke dalam perhitungan bobot

$$\begin{bmatrix} \text{coeff } a_i \\ \text{coeff } b_i \\ \text{coeff } c_i \end{bmatrix} = w_{\text{fuel}} \begin{bmatrix} \text{coeff } a_i^{\text{economic}} \\ \text{coeff } b_i^{\text{economic}} \\ \text{coeff } c_i^{\text{economic}} \end{bmatrix} + w_{\text{emission}} \left( h_i \begin{bmatrix} \text{coeff } a_i^{\text{emission}} \\ \text{coeff } b_i^{\text{emission}} \\ \text{coeff } c_i^{\text{emission}} \end{bmatrix} \right) \quad (3.9)$$

Untuk nilai *economic tendency*

$$\begin{bmatrix} \text{coeff } a_i \\ \text{coeff } b_i \\ \text{coeff } c_i \end{bmatrix} = \begin{bmatrix} \text{coeff } a_i^{\text{economic}} \\ \text{coeff } b_i^{\text{economic}} \\ \text{coeff } c_i^{\text{economic}} \end{bmatrix} \quad (3.10)$$

Untuk nilai *emission tendency*

$$\begin{bmatrix} \text{coeff } a_i \\ \text{coeff } b_i \\ \text{coeff } c_i \end{bmatrix} = \begin{bmatrix} \text{coeff } a_i^{\text{emission}} \\ \text{coeff } b_i^{\text{emission}} \\ \text{coeff } c_i^{\text{emission}} \end{bmatrix} \quad (3.11)$$

## 3.2 Lagrangian Relaxation

Pada tugas akhir ini metode yang digunakan untuk menyelesaikan permasalahan ialah *lagrangian relaxation*. Dengan *lagrangian relaxation* hasil yang didapatkan dari permasalahan *unit commitment* tidak hanya pada penjadwalan unit setiap interval tapi juga pengoptimalan pembebanan pada masing-masing unit yang beroperasi setiap interval. *Lagrangian relaxation* merupakan penyelesaian dengan dua solusi yaitu *dynamic programming* dan *lagrange multiplier*.

Metode ini didasarkan pada pendekatan dual optimisasi. Prosedur dual optimisasi berusaha untuk mendapatkan konstrain optimum dengan

memaksimalkan persamaan lagrange dengan mematuhi *lagrange multiplier* saat meminimisasi variabel lain. Metode ini dasarnya ialah menemukan nilai  $\lambda^t$  optimum masing-masing interval. Setelah unit yang aktif telah ditentukan pada *lagrangian relaxation* dan *dynamic programming* maka  $\lambda^t$  sudah dapat dicari dengan ketentuan yang berlaku. Diberikan contoh perhitungan untuk melakukan *multiobjective unit commitment* dengan *lagrangian relaxation*. Contoh perhitungan menggunakan 3 unit dengan data *cost function*, *emission function*, *spinning reserve*, minimum *up-* dan *down-time*, dan *start-up cost*.

Pertama-tama didefinisikan,

$U_i^t = 0$  untuk unit  $i$  yang *off-line* saat interval  $t$

$U_i^t = 0$  untuk unit  $i$  yang *on-line* saat interval  $t$

Dengan konstrain dan fungsi obyektif pada bab 2 dapat dibentuk fungsi lagrange dengan cara yang sama pada masalah *combined economic emission dispatch*:

$$\mathcal{L}(P, U, \lambda) = F(P_i^t, U_i^t) + \sum_{i=1}^T \lambda^t \left( P_{load}^t - \sum_{i=1}^N P_i^t U_i^t \right) \quad (3.12)$$

$$q(\lambda) = \min \mathcal{L}(P, U, \lambda) \quad (3.13)$$

$$q^*(\lambda) = \max q(\lambda) \quad (3.14)$$

Prosedur *lagrangian relaxation* menyelesaikan *unit commitment* dengan mengabaikan konstrain pada persamaan *coupling constraint*. Secara sederhana *lagrangian relaxation* dilakukan dengan dua langkah sederhana:

1. Temukan nilai  $\lambda^t$  setiap interval yang membuat nilai *duality gap* memenuhi persyaratan.
2. Diasumsikan nilai  $\lambda^t$  telah didapatkan pada langkah pertama, temukan nilai minimum  $\mathcal{L}$  dengan nilai  $P^t$  dan  $U^t$  yang telah diatur.

$$\min q(\lambda) = \sum_{i=1}^N \min \sum_{t=1}^T \{ [F_i(P_i^t + \text{Start up cost}_{i,t}) U_i^t - \lambda^t P_i^t U_i^t] \} \quad (3.15)$$



Subject to

$$U_i^t P_i^{min} \leq P_i^t \leq U_i^t P_i^{max}$$

Dan pada konstrain *up*- dan *down*-time diselesaikan dengan *dynamic programming*. Bila  $U_i^t = 0$  maka unit tersebut diabaikan, dan bila  $U_i^t = 1$  maka fungsi unit tersebut harus diminimisasi ditunjukkan pada persamaan(3.16).

$$\min[F_i(P_i) - \lambda^t P_i^t] \quad (3.16)$$

Nilai minimum dari persamaan (3.16) didapatkan dari turunan pertama persamaan (3.16).

$$\frac{d}{dP_i^t} [F_i(P_i) - \lambda^t P_i^t] = \frac{d}{dP_i^t} F_i(P_i^t) - \lambda^t = 0 \quad (3.17)$$

Dan solusi dari persamaan (3.17) adalah

$$\frac{d}{dP_i^t} F_i(P_i^{opt}) = \lambda^t \quad (3.18)$$

Untuk penyelesaian pada persamaan (3.18) terdapat tiga hal yang menentukan hasil dari  $P_i^{opt}$ , sebagai berikut :

1. Jika  $P_i^{opt} \leq P_i^{min}$ , maka

$$\min[F_i(P_i) - \lambda^t P_i^t] = F_i(P_i^{min}) - \lambda^t P_i^{min} \quad (3.19)$$

2. Jika  $P_i^{min} \leq P_i^{opt} \leq P_i^{max}$ , maka

$$\min[F_i(P_i) - \lambda^t P_i^t] = F_i(P_i^{opt}) - \lambda^t P_i^{opt} \quad (3.20)$$

3. Jika  $P_i^{max} \leq P_i^{opt}$ , maka

$$\min[F_i(P_i) - \lambda^t P_i^t] = F_i(P_i^{max}) - \lambda^t P_i^{max} \quad (3.21)$$

Perlu diketahui bahwa kita meminimisasi persamaan  $[F_i(P_i) - \lambda^t P_i^t]$  pada setiap *stage* dan pada saat  $U_i^t = 0$  maka persamaan tersebut

bernilai nol, maka cara untuk meminimisasikan persamaan tersebut adalah

$$[F_i(P_i) - \lambda^t P_i^t] < 0 \quad (3.22)$$

Dengan semua persamaan *lagrangian relaxation* yang dijelaskan diatas semua permasalahan tersebut diselesaikan dengan nilai lambda yang telah didapatkan untuk penyelesaian permasalahan yang optimal. Pada *lagrangian relaxation* nilai lambda dicari dengan menggunakan *dual optimization*.

$$\lambda^t = \lambda^t + \left[ \frac{d}{d\lambda} q(\lambda) \right] \alpha \quad (3.23)$$

Setiap pencarian  $\lambda^t$  dilakukan secara terpisah. Nilai  $\alpha$  pada pencarian lambda harus ditentukan dengan benar agar nilai lambda mencapai nilai optimalnya. Jadi, apabila total daya yang dibangkitkan  $P_i^t$  tidak memenuhi *demand load* atau  $P_{load}^t$  dengan selisih nilai kurang dari 0 maka nilai lambda akan menggeser ke kanan menuju ke *stationery point*. Lalu apabila total daya yang dibangkitkan  $P_i^t$  tidak memenuhi *demand load* atau  $P_{load}^t$  dengan selisih nilai lebih dari 0 maka nilai lambda akan menggeser ke kiri mendekati *stationery point*.

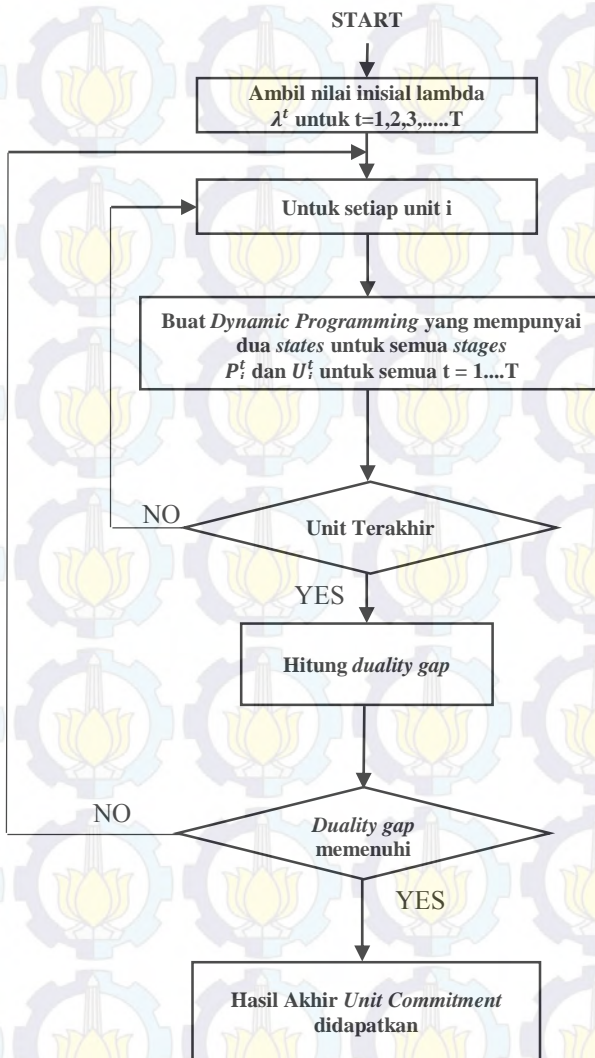
Di unit commitment metode *lagrangian relaxation* menggunakan iterasi dari algoritma *dynamic programming* untuk mendapatkan nilai awal yang baik untuk melakukan iterasi dual optimisasi. Dan pada akhir solusi *lagrangian relaxation*, logika *dynamic programming* akan mendapatkan solusi akhir dari permasalahan *unit commitment* menggunakan *lagrangian relaxation*.

Dimana lambda vektor akan bergeser ke kanan dan ke kiri, dengan syarat persamaan (3.24) dan persamaan (3.25)

$$\alpha = 0.01 \text{ ketika } \frac{d}{d\lambda} q(\lambda) \text{ bernilai positif} \quad (3.24)$$

$$\alpha = 0.002 \text{ ketika } \frac{d}{d\lambda} q(\lambda) \text{ bernilai negatif} \quad (3.25)$$

Dengan setiap interval nilai lambda akan diperlakukan secara terpisah.



**Gambar 3.2** Diagram alur *lagrangian relaxation*



### 3.3 Argumen *Input Multiobjective Unit Commitment*

Dari semua data yang diolah untuk mendapatkan hasil *unit commitment* pada *software* Delphi, data *input* tersebut akan dibuat kedalam argumen. Nantinya argumen *input* tersebut akan dimasukkan kedalam perhitungan. Argumen *input* yang digunakan dapat dilihat sebagai berikut:

**Tabel 3.1** Argumen *Input*

Argumen	Keterangan
CoeffEconomic [i]	Sebagai inputan awal dari nilai coefisien biaya a, b, c dalam persamaan $H_i = a_i P_i^2 + b_i P_i + c_i$
CoeffEmission [i]	Sebagai inputan awal dari nilai coefisien emisi $\alpha$ , $\beta$ , $\gamma$ dalam persamaan $E_i = \alpha_i P_i^2 + \beta_i P_i + \gamma_i$
Unitmax[i]	Sebagai inputan awal dari batas maximum pembangkitan unit (Pmax)
Unitmin[i]	Sebagai inputan awal dari batasan minimum pembangkitan unit (Pmin)
Fuelcost[i]	Sebagai inputan awal nilai dari fuelcost yang digunakan untuk persamaan $Fi(Pi(t)) = Hi(Pi(t)) \times fuelcost\ i$
Minimum up-time[i]	Sebagai inputan untuk batasan setiap unit i mempunyai waktu nyala minimal.
Minimum down-time[i]	Sebagai inputan untuk batasan setiap unit i mempunyai rentang waktu tidak boleh dinyalakan kembali.
SPR[i]	Sebagai inputan <i>spinning reserve</i> pada setiap unit i untuk semua interval $P_{i\ min}^t \leq P_{i\ generate}^t + SPR_i \leq P_{i\ max}^t$
Number of Interval	Jumlah interval yang akan dimasukkan kedalam perhitungan <i>unit commitment</i> .
Load Interval	Sebagai inputan nilai beban ( <i>Pload</i> ) di setiap interval.

### 3.4 Sintaksis *Software Multiobjective Unit Commitment* dengan Delphi

Sintaksis *software* adalah perintah yang digunakan untuk melakukan pemanggilan program dengan argumen input yang kita masukkan. Sintaksis yang digunakan pada *software* Delphi adalah sebagai berikut :

**Tabel 3.2** Sintaksis

Sintaksis	Keterangan
datadump	Digunakan sebagai memasukkan semua data awal perhitungan di setiap periode
Sort	Merupakan prosedur untuk mengurutkan nilai <i>price penalty factor</i> dari yang terkecil hingga terbesar.
ihr_ftn	Sebagai inisiasi turunan pertama persamaan $\varphi_c(P_i)$
EDC	Sebagai prosedur untuk mencari nilai $P_i^{opt}$ untuk setiap nilai lambda yang didapatkan.
lambda_search	Sebagai prosedur penjalanan menentukan nilai lambda yang digunakan dalam persamaan, yaitu dengan menentukan nilai $\lambda_{start}$ dan $\frac{d}{d\lambda}q(\lambda)$ untuk proses iterasi. Hasil akhir bertujuan untuk mendapatkan jumlah nilai <i>duality gap</i> yang bernilai $\leq 0.005$
FEAS	Prosedur untuk menentukan unit i aktif atau tidak untuk semua <i>stage</i>
BESTSTATE	Sebagai prosedur penentuan apakah unit tersebut merupakan <i>state</i> terbaik dengan memperhatikan <i>start-up cost</i> dan minimum <i>up- and down-time</i>
prod_cost	Mendapatkan nilai biaya pembangkitan setiap unit setelah mendapatkan nilai pembangkitan yang optimal dari proses <i>lagrangian relaxation</i>
Final Output	Prosedur <i>print</i> untuk menampilkan hasil akhir dari <i>multiobjective unit commitment</i> .

### 3.5 Penggunaan *software* untuk perhitungan *Multiobjective Unit Commitment*.

The screenshot shows a software window titled "NewUnitComFile.Dat" with a menu bar (File) and a title bar. The main area is titled "UNIT COMMITMENT" and "Data Generator". It features a table with columns: Unit Number, Min MW, Max MW, EC1, EC2, EC3, EM2, EM3, and Start Up Cost. Below the table, there are input fields for "Number Of Units" (set to 0) and "Number Of Intervals". There are also radio buttons for "Spinning Reserve per Unit" (Yes/No) and "Calculate Weight" (EconomicDispatch, EmissionDispatch, CombinedEmissionD, CombinedEconomicD). A "Process" button is at the bottom right.

**Gambar 3.3** *Main Display Multiobjective Unit Commitment*

Gambar 3.3 menunjukkan tampilan utama dari *software* perhitungan *multiobjective unitcommitment*. Pada *windows* utama MUC, data *input* dimasukkan kedalam masing-masing *subject*. Pada kolom data generator terdapat data *input* setiap unit dengan berdasarkan jumlah unit yang diminta. Untuk memasukkan data input dibutuhkan media untuk tempat memasukkan input. Jika pada data masukkan di klik *double* maka akan muncul *window* baru yang akan menuju *editor input* tiap unit untuk memasukkan data *input*. Gambar 3.4 merupakan tampilan yang digunakan untuk memasukkan data input.

Terdapat juga kolom untuk memasukkan data beban maksimal untuk 24 interval, kolom ditunjukkan pada gambar 3.5.



Unit Number	1
Min (MW)	0.0
Max (MW)	0.0
StartUpCost	0.0
Fuel Cost	0.0
Min Up Time	0
Min Down Time	0
Initial Status	0
EC1	0.00000
EC2	0.00000
EC3	0.00000
EM1	0.00000
EM2	0.00000
EM3	0.00000

**Gambar 3.4** Editor Input

	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5

**Gambar 3.5** Input untuk beban setiap interval

Gambar 3.6 merupakan kolom *input* untuk memasukkan data beban setiap *interval*. Jumlah data beban yang diberikan sesuai dengan jumlah *interval* yang dimasukkan, dengan batasan maksimal sebanyak 24 *interval*.

Calculate Weight

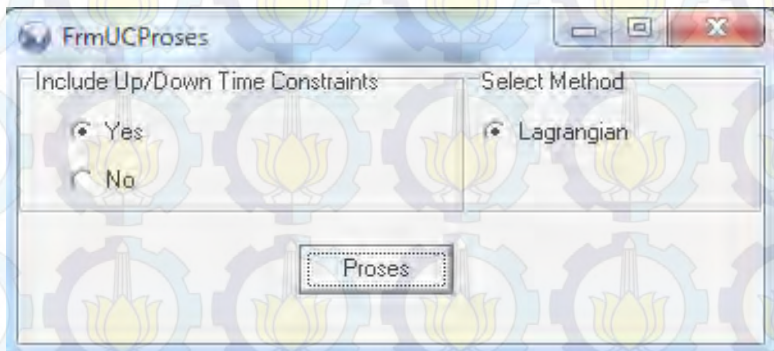
☒ EconomicDispatch
 ☐ CombinedEED
 ☐ EmissionDispatch

**Gambar 3.6** Weight Preferences Option perhitungan pada *software*

Pada kolom *calculate weight* akan ditentukan hasil *unit commitment* akan lebih cenderung ke sisi biaya atau sisi emisi.

1. EconomicDispatch  
Bobot akan ditaruh seluruhnya pada fungsi biaya.
2. EmissionDispatch  
Bobot akan ditaruh seluruhnya pada fungsi emisi.
3. CombinedEED  
Menggunakan *price penalty factor* untuk *composite* kedua fungsi menjadi *single objective function*. Dan pada *option* ini *price penalty factor* diletakkan pada fungsi emisi

Gambar 3.7 menunjukan metode yang akan ingin dipilih untuk penyelesaian *unit commitment* dan *option* untuk memilih memperhitungkan minimum *up- and down-time* atau tidak.



**Gambar 3.7** Display untuk *min.up- dan down-time option* dan *method options*





## BAB 4 HASIL DAN SIMULASI

Pada bab ini akan diperlihatkan hasil perhitungan *multiobjective unit commitment* dengan menggunakan *lagrangian relaxation* dengan hasil unit “on” dan “off” dan juga disertakan hasil pembebanan setiap unit. Hasil perhitungan yang diperlihatkan berupa tabel-tabel yang meliputi pembangkitan yang optimal, biaya total, emisi total dan penjadwalan pembangkit. Hasil dari simulasi ini akan memperlihatkan perbandingan dari aplikasi metode untuk beberapa kasus. Data yang dimasukkan ke dalam perhitungan adalah data IEEE 39 bus. Berikut merupakan data yang dimasukkan dalam simulasi :

**Tabel 4.1** Data masukan *cost function* IEEE 39 bus

Unit	Pmin (MW)	Pmax (MW)	$a_i$ (\$/MWh <sup>2</sup> )	$b_i$ (\$/MWh)	$c_i$ (\$)
1	150	455	0,0004	16,19	1000
2	150	455	0,0003	17,26	970
3	20	130	0,002	16,6	700
4	20	130	0,0021	16,5	680
5	25	162	0,0039	19,7	450
6	20	80	0,0071	22,36	370
7	25	85	0,0079	27,74	480
8	10	55	0,0022	27,27	665
9	10	55	0,0017	27,79	670
10	10	55	0,0041	25,92	660

**Tabel 4.2** Data masukan *emission function* IEEE 39 bus

Unit	$\alpha$ (ton/MWh <sup>2</sup> )	$\beta$ (ton/MWh)	$\gamma$ (ton)
1	0,00312	-0,24444	10,33908
2	0,00312	-0,24444	10,33908
3	0,00509	-0,40695	30,0391
4	0,00509	-0,40695	30,0391

**Tabel 4.2** Data masukan *emission function* IEEE 39 bus (lanjutan)

Unit	$\alpha(\text{ton/MWh}^2)$	$\beta(\text{ton/MWh})$	$\gamma(\text{ton})$
5	0,00344	-0,38132	32,00006
6	0,00344	-0,38132	32,00006
7	0,00465	-0,39023	33,00056
8	0,00465	-0,39254	35,00056
9	0,0047	-0,39864	36,00012
10	0,00465	-0,39023	33,00066

**Tabel 4.3** Data masukan minimum *up- and down-time, start-up cost, dan* inisial status

Unit	Up-Time (h)	Down Time(h)	Start-up Cost (\$)	Initial Status (h)
1	8	8	4500	8
2	8	8	5000	8
3	5	5	550	-5
4	5	5	560	-5
5	6	6	900	-6
6	3	3	170	-3
7	3	3	260	-3
8	1	1	30	-1
9	1	1	30	-1
10	1	1	30	-1

## 4.1 Studi Kasus

### 4.1.1 Kasus 1

Pada studi kasus 1 diperlihatkan hasil perhitungan yang hanya memperhatikan dari sisi biaya bahan bakar BEC(*Best Cost Commitment*). Pada BEC  $w_{fuel} = 1$  dan  $w_{emission} = 0$ , maka  $\varphi_c(P_i) = F_c(P_i)$  dimana  $F_c$  merupakan *cost function*.

**Tabel 4.4** Hasil *running* aplikasi *PowerGen* (BCC) Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
4	OFF	OFF	OFF	ON	ON	ON	ON	ON
5	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
6	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
8	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
9	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
10	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
Cost(\$)	13687	14527	16302	18590	19473	21738	22856	23901
Emission(ton)	684,3	753,9	947,9	941,6	1052	1175	1185	1212
Load(MW)	700	750	850	950	1000	1100	1150	1200

**Tabel 4.5** Hasil *running* aplikasi *PowerGen* (BCC) Interval 9-16

Unit \ Interval	9	10	11	12	13	14	15	16
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	OFF	OFF
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	ON	ON
6	OFF	ON	ON	ON	ON	OFF	OFF	OFF
7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
8	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF
9	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
10	OFF	OFF	ON	ON	OFF	OFF	OFF	OFF
Cost(\$)	26161	28752	30686	32600	28752	26161	23901	20847



**Tabel 4.5** Hasil *running* aplikasi *PowerGen*(BCC) Interval 9-16 (lanjutan)

Unit \ Interval	9	10	11	12	13	14	15	16
Emission(ton)	1257	1299	1325	1351	1299	1257	1212	1135
Load(MW)	1300	1400	1450	1500	1400	1300	1200	1050

**Tabel 4.6** Hasil *running* aplikasi *PowerGen* (BCC) Interval 17-24

Unit \ Interval	17	18	19	20	21	22	23	24
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	OFF	OFF	OFF	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	OFF	OFF	OFF
5	ON	ON	ON	ON	ON	OFF	OFF	OFF
6	OFF	OFF	OFF	ON	ON	ON	OFF	OFF
7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
8	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
9	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
10	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
Cost(\$)	19998	21838	23901	28752	26574	21964	17782	16027
Emission(ton)	1022	1175	1212	1299	1274	1175	851,5	709,7
Load(MW)	1000	1100	1200	1400	1300	1100	900	800

Total biaya pada kasus 1 adalah sebesar 548930,96 \$ dan total emisi yang dihasilkan pada semua interval sebesar 26804,43 Ton.

#### 4.1.2 Kasus 2

Pada studi kasus 1 diperlihatkan hasil perhitungan yang hanya memperhatikan dari sisi ekonomi BEC(*Best Economic Commitment*). Pada BEC  $w_{fuel} = 0$  dan  $w_{emission} = 1$ , maka  $\varphi_c(P_i) = E_c(P_i)$  dimana  $E_c$  merupakan *emission function*.

**Tabel 4.7** Hasil *running* aplikasi *PowerGen* (BEC) Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
1	ON	ON	ON	ON	ON	ON	ON	ON

**Tabel 4.7** Hasil *running* aplikasi *PowerGen* (BEC) Interval 1-8 (lanjutan)

Unit \ Interval	1	2	3	4	5	6	7	8
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	ON	ON
6	ON	ON	ON	ON	ON	ON	ON	ON
7	ON	ON	ON	ON	ON	ON	ON	ON
8	OFF	OFF	ON	ON	ON	ON	ON	ON
9	OFF	OFF	OFF	ON	ON	ON	ON	ON
10	OFF	OFF	ON	ON	ON	ON	ON	ON
Cost(\$)	18088	19100	23189	26235	27161	28888	29717	30577
Emission(ton)	223,9	242	289	339	370	448	493	548
Load(MW)	700	750	850	950	1000	1100	1150	1200

**Tabel 4.8** Hasil *running* aplikasi *PowerGen* (BEC) Interval 9-16

Unit \ Interval	9	10	11	12	13	14	15	16
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	ON	ON
6	ON	ON	ON	ON	ON	ON	ON	ON
7	ON	ON	ON	ON	ON	ON	ON	ON
8	ON	ON	ON	ON	ON	ON	ON	ON
9	ON	ON	ON	ON	ON	ON	ON	ON
10	ON	ON	ON	ON	ON	ON	ON	ON
Cost(\$)	32255	33952	34803	35661	33952	32255	30577	28025
Emission(ton)	678	841	934	1.035	840,6	678	548	406

**Tabel 4.8** Hasil *running* aplikasi *PowerGen* (BEC) Interval 17-24 (lanjutan)

Unit \ Interval	9	10	11	12	13	14	15	16
Load(MW)	1300	1400	1450	1500	1400	1300	1200	1050

**Tabel 4.9** Hasil *running* aplikasi *PowerGen* (BEC) Interval 17-24

Unit \ Interval	17	18	19	20	21	22	23	24
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	OFF
4	ON	ON	ON	ON	ON	ON	ON	OFF
5	ON	ON	ON	ON	ON	ON	ON	OFF
6	ON	ON	ON	ON	ON	ON	ON	OFF
7	ON	ON	ON	ON	ON	ON	ON	OFF
8	ON	ON	ON	ON	ON	ON	ON	OFF
9	ON	ON	ON	ON	ON	ON	OFF	OFF
10	ON	ON	ON	ON	ON	ON	ON	OFF
Cost(\$)	27161	28888	30577	33952	32255	28888	24126	15466
Emission(ton)	370	448	548	841	678	448	314	824
Load(MW)	1000	1100	1200	1400	1300	1100	900	800

Total biaya pada kasus 2 adalah sebesar 688279,32 \$ dan total emisi yang dihasilkan pada semua interval sebesar 13383,21 Ton.

#### 4.1.3 Kasus 3

Pada studi kasus 3 diperlihatkan hasil perhitungan yang memperhatikan dua fungsi, fungsi biaya dan fungsi emisi, yang dibentuk kedalam *single objective function*.

**Tabel 4.10** Hasil *running* aplikasi *PowerGen* (*Combined Economic Emission*) Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON



**Tabel 4.10** Hasil *running* aplikasi *PowerGen(Combined Economic Emission)* Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
3	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
5	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
6	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON
7	OFF	OFF	OFF	OFF	ON	ON	ON	ON
8	OFF	OFF	ON	ON	ON	ON	ON	ON
9	OFF	ON	ON	ON	ON	ON	ON	ON
10	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON
Cost(\$)	13759	15877	18813	21675	22812	25360	26047	28224
Emission(ton)	614	633	748	869	922	991	1.031	1.125
Load(MW)	700	750	850	950	1000	1100	1150	1200

**Tabel 4.11** Hasil *running* aplikasi *PowerGen (Combined Economic Emission)* Interval 9-16

Unit \ Interval	9	10	11	12	13	14	15	16
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	OFF	OFF	ON	ON	OFF	ON	OFF
4	ON	OFF	ON	ON	OFF	OFF	OFF	OFF
5	OFF	ON	ON	ON	ON	ON	OFF	OFF
6	OFF	ON	ON	OFF	ON	ON	OFF	OFF
7	ON	ON	ON	ON	OFF	ON	ON	ON
8	ON	ON	ON	ON	ON	ON	ON	ON
9	ON	ON	ON	ON	ON	ON	ON	ON
10	OFF	ON	OFF	OFF	ON	OFF	ON	ON
Cost(\$)	29249	32616	32952	33640	31846	29751	28061	24828
Emission(ton)	1.141	1.282	1.255	1.295	1.198	1.143	1.045	936
Load(MW)	1300	1400	1450	1500	1400	1300	1200	1050

**Tabel 4.12** Hasil *running* aplikasi *PowerGen (Combined Economic Emission)* Interval 17-24

Unit \ Interval	17	18	19	20	21	22	23	24
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
4	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF
5	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF
6	ON	OFF	OFF	OFF	ON	OFF	OFF	OFF
7	ON	ON	ON	ON	ON	ON	OFF	OFF
8	ON	OFF	ON	ON	ON	ON	ON	OFF
9	OFF	ON	ON	ON	ON	ON	ON	ON
10	OFF	ON	ON	ON	ON	OFF	ON	ON
Cost(\$)	22388	25124	28029	32451	30574	25197	20826	17891
Emission (ton)	862	918,7	1045	1196	1115	918,5	763,2	648,8
Load(MW)	1000	1100	1200	1400	1300	1100	900	800

Total biaya pada kasus 3 adalah sebesar 627747,85 \$ dan total emisi yang dihasilkan pada semua interval sebesar 23695,47 Ton.

#### 4.1.4 Kasus 4

*Spinning Reserve* atau cadangan berputar merupakan hal yang perlu diperhatikan untuk menunjang keandalan kontinuitas sistem tenaga listrik. Pada kasus 4 ini diasumsikan pada masing-masing generator untuk memiliki *spinning reserve* agar sistem tetap beroperasi apabila terdapat salah satu unit yang *collapse*. Pada kasus 4 sistem akan diberikan *spinning reserve* sebesar daya maksimal salah satu unit yang terbesar, yaitu 455 MW, dan *spinning reserve* akan disebar secara merata pada seluruh unit dari kapasitas maksimal daya pembangkitan seluruh unit.

$$\begin{aligned}
 \text{SPR} &= \frac{455}{\sum_{i=1}^N P_i^{\max}} \times 100\% \\
 &= \frac{455}{1662} \times 100\% \text{ MW} \\
 &= 27,3 \% \text{ dari total seluruh pembangkitan}
 \end{aligned}$$

**Tabel 4.13** Hasil *running* aplikasi *PowerGen (Combined Economic Emission with SPR)* Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	OFF	OFF	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	OFF	OFF	OFF	ON	ON	ON	ON	ON
6	OFF	OFF	OFF	OFF	ON	ON	ON	ON
7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
8	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
9	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
10	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
Cost(\$)	14416	15265	17645	20170	21728	25532	27488	29515
Emission(ton)	481,3	567,3	613,8	616,2	623	707,2	752,5	757,5
Load(MW)	700	750	850	950	1000	1100	1150	1200

**Tabel 4.14** Hasil *running* aplikasi *PowerGen (Combined Economic Emission with SPR)* Interval 9-16

Unit \ Interval	9	10	11	12	13	14	15	16
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	ON	ON
6	ON	ON	ON	ON	ON	ON	ON	ON
7	ON	ON	ON	ON	ON	ON	ON	OFF
8	ON	ON	ON	ON	ON	ON	ON	OFF
9	ON	ON	ON	ON	ON	ON	ON	OFF
10	ON	ON	ON	ON	ON	ON	ON	ON
Cost(\$)	29515	29515	29515	29515	29515	29515	29515	23602
Emission(ton)	757,5	757,5	757,5	757,5	757,5	757,5	757,5	665



**Tabel 4.14** Hasil *running* aplikasi *PowerGen(Combined Economic Emission with SPR)* Interval 9-16 (lanjutan)

Unit \ Interval	9	10	11	12	13	14	15	16
Load(MW)	1200	1200	1200	1200	1200	1200	1200	1050

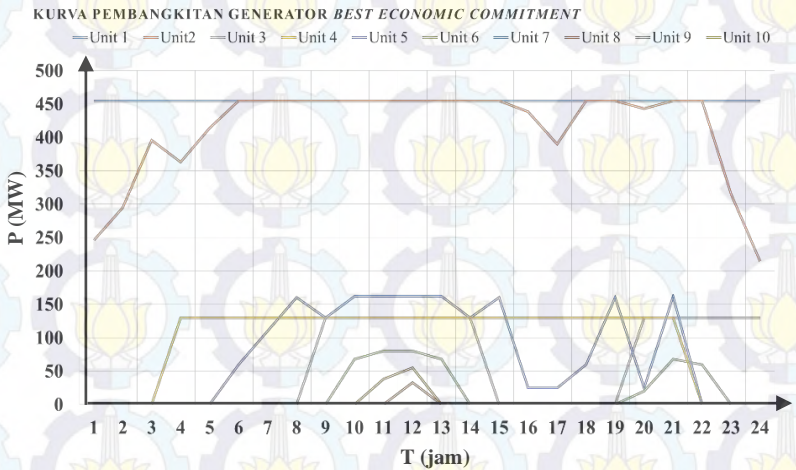
**Tabel 4.15** Hasil *running* aplikasi *PowerGen (Combined Economic Emission with SPR)* Interval 17-24

Unit \ Interval	17	18	19	20	21	22	23	24
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	OFF	OFF
6	ON	ON	ON	ON	ON	ON	ON	OFF
7	OFF	OFF	ON	ON	ON	OFF	OFF	OFF
8	OFF	ON	ON	ON	ON	ON	OFF	OFF
9	OFF	OFF	ON	ON	ON	OFF	OFF	OFF
10	OFF	ON	ON	ON	ON	ON	OFF	OFF
Cost(\$)	21728	25532	29515	29515	29515	25532	19203	16798
Emission(ton)	623	707,2	757,5	757,5	757,5	707,2	620,1	526,9
Load(MW)	1000	1100	1200	1200	1200	1100	900	800

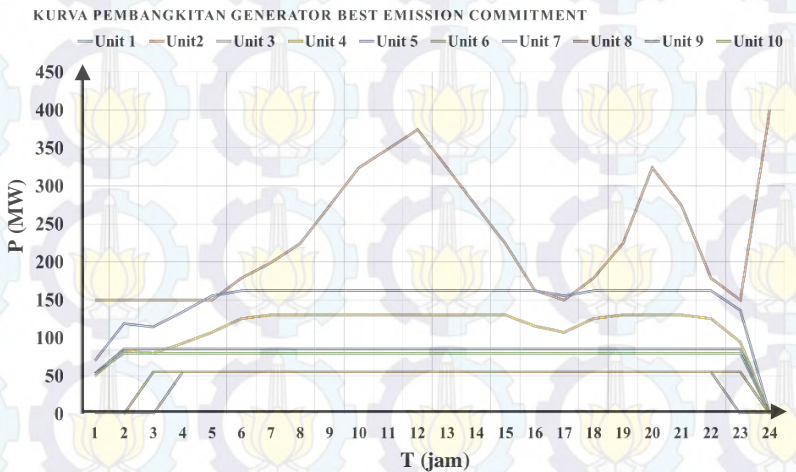
Dapat dilihat pada Tabel 4.14 untuk interval ke-9 hingga 15, interval 20 dan interval 21 melewati batasan maksimum dari kapasitas daya maksimum 10 unit pembangkit setelah diberi masukkan *spinning reserve*, maka untuk pembangkitan yang lebih besar dari 1200 MW seluruh unit tidak dapat beroperasi. Maka untuk semua rentang waktu pembebanan sistem beroperasi hanya pada interval 1 hingga 13 hingga 24. Salah satu solusinya adalah melakukan pemangkasan beban sehingga semua beban setiap interval yang melewati batasan maksimum akan dipangkas hingga 1200 MW. Dan total biaya seluruh operasi untuk 24 interval pada kasus 4 adalah sebesar 602184,28 \$ dan emisi yang dihasilkan sebesar 16542,66 ton.

Seperti pada kasus 4 akan diberikan hasil dari interval beban seperti pada kasus 4 tetapi *spinning reserve* tidak dimasukkan kedalam perhitungan.

## 4.2 Kurva Daya untuk semua kasus.



Gambar 4.1 Kurva daya *Best Economic Commitment*



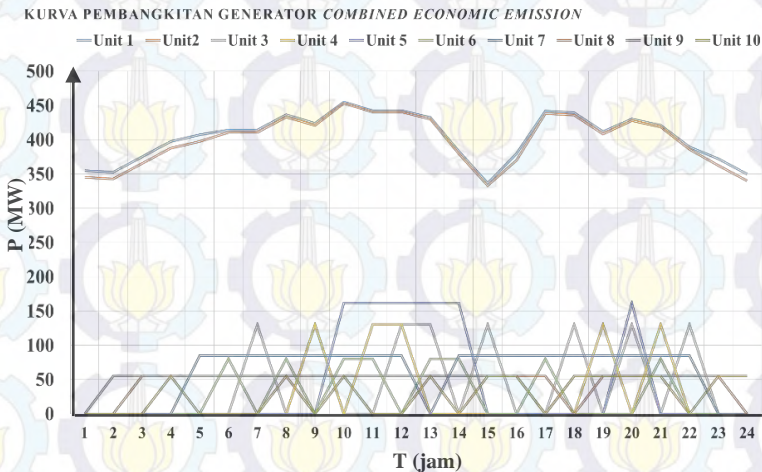
Gambar 4.2 Kurva daya *Best Emission Commitment*

Gambar 4.1 merupakan kurva daya terbangkit setiap generator pada 24 interval dengan rentang interval setiap 1 jam dengan pengoptimalan hanya pada fungsi biaya. Dan terlihat bahwa unit 1 dan unit 2 lebih

cenderung terbebani dengan beban yang lebih besar dikarenakan fungsi biaya unit dan unit 2 merupakan fungsi biaya yang lebih minimal daripada unit lainnya. Sehingga hasil yang didapatkan lebih cenderung ke sisi biaya tanpa memperhatikan emisi yang dihasilkan.

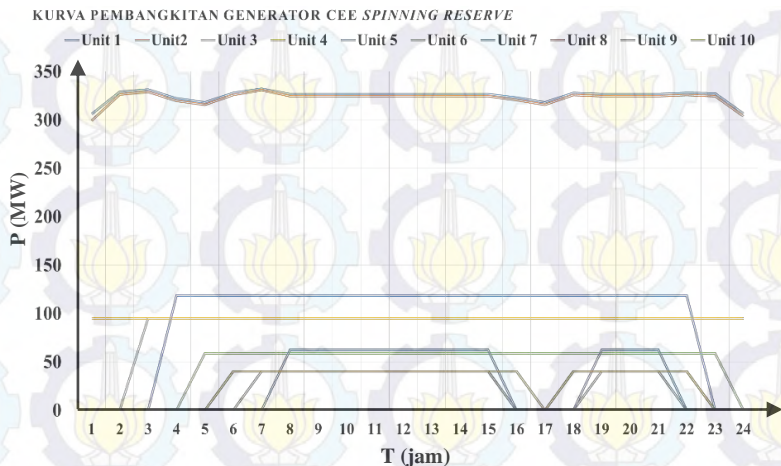
Pada Gambar 4.2 merupakan kurva daya terbangkit dengan hanya mengoptimalkan fungsi emisi. Pada gambar terlihat unit 1 dan unit 2 tidak dibebankan secara maksimal sedangkan pada unit lain ditekankan pada kapasitas daya maksimal. Sehingga didapatkan emisi yang paling optimal tetapi tidak memperhatikan dari sisi biaya.

Untuk memperhatikan biaya dan emisi yang dihasilkan maka diperlukannya *composite* kedua fungsi, fungsi biaya dan fungsi emisi. Kedua fungsi tersebut dibentuk kedalam *single objective function*, dan daya yang dibangkitkan setiap unit terlihat pada Gambar 4.3. hasil yang didapatkan biaya pada *combined economic emission* lebih mahal daripada *economic dispatch* tetapi lebih murah daripada *emission dispatch*. Sebaliknya emisi yang dihasilkan untuk pengoptimalan *combined economic emission* lebih besar daripada *emission dispatch* tetapi hasil emisinya lebih sedikit bila dibandingkan dengan *economic dispatch*.



**Gambar 4.3** Kurva daya *Combined Economic Emission*

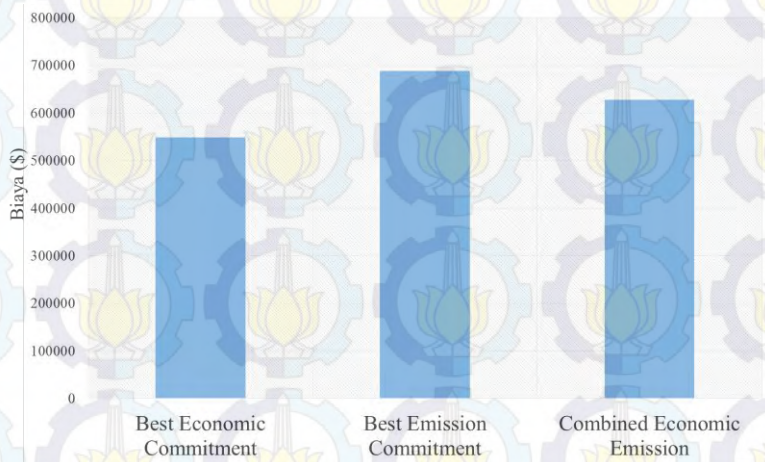




**Gambar 4.4** Kurva daya *Combined Economic Emission with Spinning Reserve*

### 4.3 Analisa Semua Kasus

Kasus pertama dan kedua dengan masing-masing memiliki kecenderungan ke sisi biaya bahan bakar dan sisi emisi, sehingga kedua kasus tersebut memiliki hasil yang cenderung untuk masing-masing sektor. Gambar 4.5 menunjukkan masing-masing hasil dari biaya yang dikeluarkan untuk memenuhi beban.

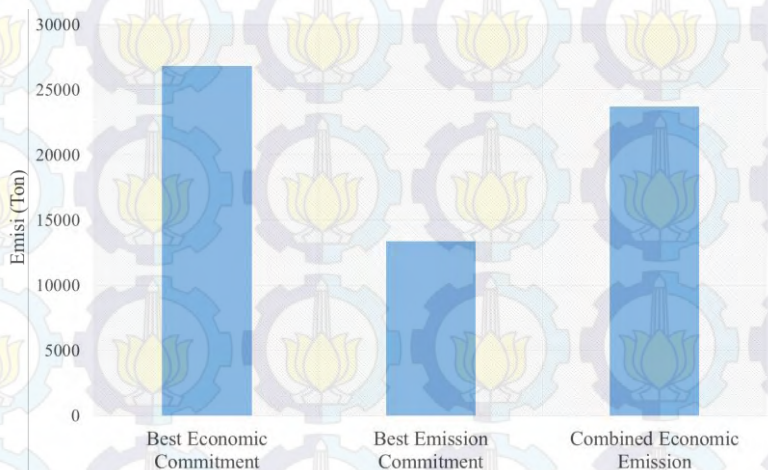


**Gambar 4.5** Grafik biaya untuk kasus 1, 2 dan 3

Biaya(BEC) = 548930,96 \$  
 Biaya(BEMC) = 688279,32 \$  
 Margin Biaya (Biaya-Emisi) = -139349,22 \$

Biaya pada kasus pertama lebih murah dikarenakan fungsi biaya untuk pembangkit dengan kapasitas yang terbesar, unit 1 dan unit 2, memiliki fungsi biaya yang paling minimal diantara semua pembangkit yang lain maka kedua unit tersebut selalu dibangkitkan secara maksimal bila mungkin untuk memenuhi *demand load*. Dan biaya *start-up cost* unit 1 dan unit 2 merupakan biaya yang terbesar diantara semua unit jika ditotal biaya *start-up cost* kedua unit tersebut sebesar 9500 \$, dibandingkan saja total biaya *start-up cost* delapan unit lain hanya sebesar 2500 \$ sehingga kedua unit tersebut sangat rugi bila di “off” dan “on” pada 24 interval tersebut mengingat *minimum-up* dan *down time* nya selama 8 jam. Sedangkan pada emisi unit 1 dan unit 2 menghasilkan emisi terbesar bila dibandingkan dengan unit lain untuk pembangkitan maksimal maka unit 1 dan unit 2 hanya dinyalakan untuk memenuhi beban.

Emisi(BEC) = 26804.43 Ton  
 Emisi(BEMC) = 13383,21 Ton  
 Margin Emisi (Biaya-Emisi) = 13421,22 Ton



**Gambar 4.6** Grafik emisi untuk kasus 1, 2 dan 3

Lain halnya pada emisi karena untuk *Best Emission Commitment* fungsi emisi pada unit 1 dan unit 2 bertolak belakang dengan fungsi biayanya dan oleh karena itu pada unit 1 dan unit 2 tidak pernah di setel ke kapasitas maksimumnya bila *demand load* nya sudah terpenuhi. Tetapi pada kasus 2, mengingat *minimum-up* dan *down time* yang tinggi untuk unit 1 dan unit 2 dan *demand load* yang harus dipenuhi maka unit 1 dan unit 2 selalu dinyalakan. Sehingga terlihat pada Gambar 4.6 nilai emisi yang dihasilkan pada kasus 1 sangatlah tinggi bila dibandingkan dengan kasus 2.

Pada kasus 3 mengombinasikan kedua fungsi, fungsi biaya dan fungsi emisi membuat kedua sektor biaya dan emisi tidak tindih sebelah.

$$\text{Biaya(Combined)} = 627747,85 \$$$

$$\text{Emisi(Combined)} = 23695,47 \text{ Ton}$$

$$\text{Margin Biaya(Combined-BEC)} = 78816,89 \$$$

$$\text{Margin Emisi(Combined-BEC)} = 3108,96 \text{ Ton}$$

$$\text{Margin Biaya(Combined-BEmC)} = 60531,47 \$$$

$$\text{Margin Emisi(Combined-BEmC)} = 10312,26 \text{ Ton}$$

Untuk kasus 4, menggunakan *spinning reserve* sebagai tambahan konstrain, mengakibatkan melonjaknya biaya pembangkitan dan emisi yang dihasilkan, dikarenakan untuk nilai yang dicapai belum mencapai optimal tetapi pembangkitan sudah pada batas maksimumnya akibat *spinning reserve*.

Dengan *Spinning Reserve*

$$\text{Cost} = 602184,28 \$$$

$$\text{Emission} = 16542,66 \text{ ton}$$

Tanpa *Spinning Reserve*

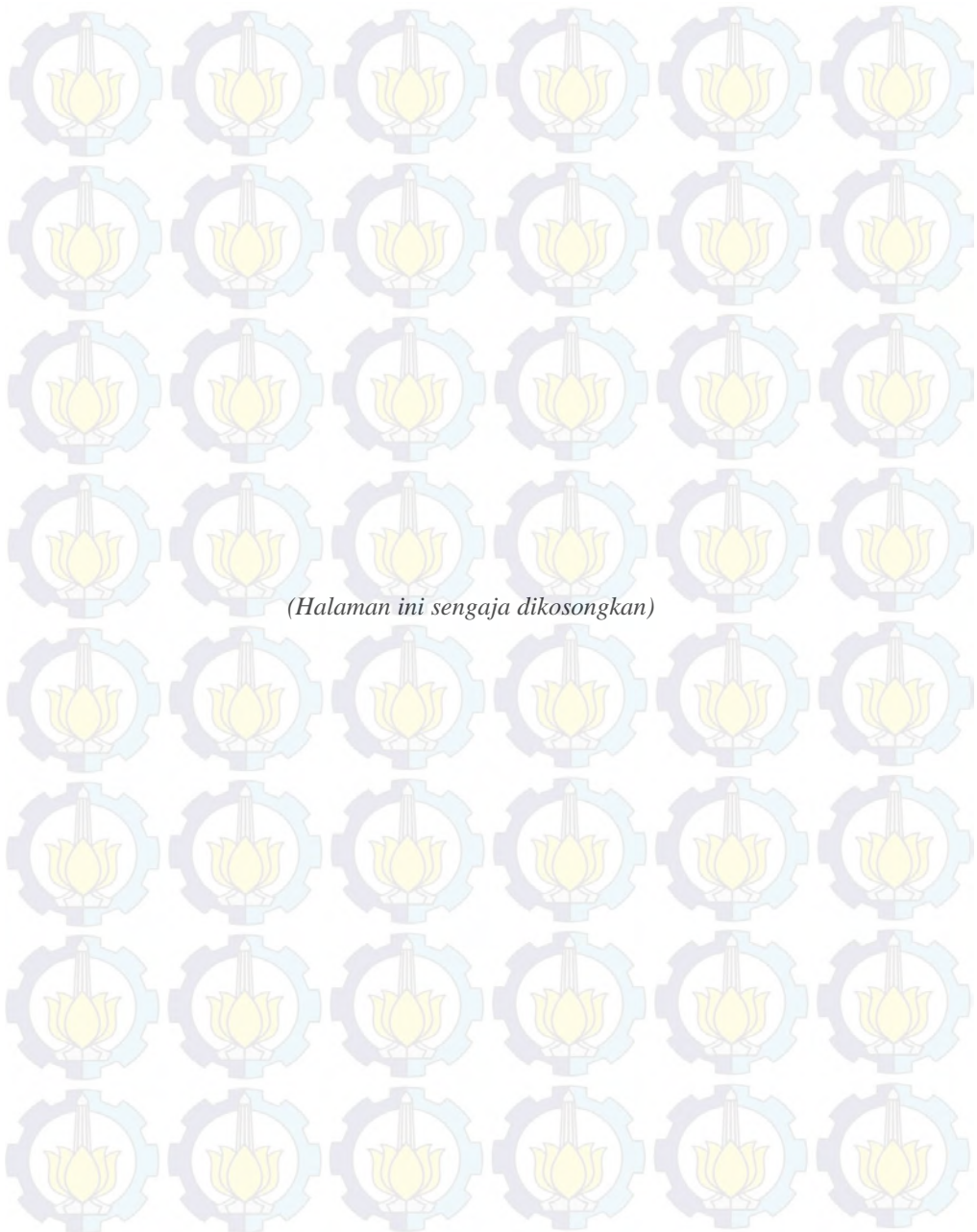
$$\text{Cost} = 573510,30 \$$$

$$\text{Emission} = 22180,34 \text{ ton}$$

$$\begin{aligned} \text{Margin Cost} &= 602184,28 - 573510,30 \\ &= 28674,28 \$ \end{aligned}$$

$$\begin{aligned} \text{Margin Emission} &= 16542,66 - 22180,34 \\ &= -5637,68 \text{ ton} \end{aligned}$$





*(Halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

- [1] Wood, Allen J & Wollenberg, Bruce F, "*Power Generation, Operation, and Control*", John Wiley & Sons Inc, 1996.
- [2] NP, Padhy, "*Unit commitment – a bibliographical survey*". IEEE Trans Power System 2004;19(2):1196–205.
- [3] Raglend, I Jacob, "*Solutions to Practical Unit Commitment Problems with Operational, Power Flow and Enviromental Constraints*", 1-4244-0493-2/06 2006 IEEE
- [4] Sundaram, C Shanmuga, "*Solution to Profit Based Unit Commitment using Swarm Intelligence Technique*", 2014 ICCPEIC
- [5] Mariano, Silvio, "*Unit Commitment with Environmental Considerations: A Practical Approach*", 15<sup>th</sup> PSCC 2005
- [6] Mandal, K K, "*Non-Convex Emission Constrained Economic Dispatch Using Self-Adaptive Particle Swarm Optimization Technique*", 1568-4946 2014 Elsevier
- [7] W. L. Snyder, H. D. Powell, J. C. Payburn, "*Dynamic Programming Approach to Unit Commitment*," IEEE Transactions on Power Systems, vol. 2, No.2, pp. 339~350,1987
- [8] Benhamida, Farid, "*Enhanced lagrangian Relaxation Solution to The Generation Scheduling Problem*", Electrical Power and Energy Systems 32 (2010) 1099-1105
- [9] Ali Ahmadian, Mahdi Sedghi, Masoud Aliakbar Golkar, "*Solving Multiobjective Unit Commitment Problem to Minimize Operation Cost and Emissions Using HBMO Algorithm*", ICMLEME 2014

## BAB 5

### KESIMPULAN & SARAN

#### 5.1 Kesimpulan

Dari hasil analisis *software* PowerGen pada *economic dispatch*, *emission dispatch* dan *combined economic emission* untuk perhitungan *multiobjective unit commitment* menggunakan *lagrangian relaxation* pada sistem kelistrikan:

1. Aplikasi perhitungan ini, PowerGen, mampu menentukan pembangkit yang harus dioperasikan dan tidak dioperasikan untuk mendapatkan pembebanan yang optimal setiap unit pembangkit.
2. Hasil yang diperoleh dari *software* PowerGen ini dapat memilih jenis pengoptimalan yang diinginkan, mulai dari sisi ekonomi, sisi emisi ataupun keduanya.
3. Hasil penjadwalan menggunakan *software* PowerGen belum menjamin untuk dapat diterapkan pada sistem interkoneksi karena adanya batasan-batasan yang belum dipenuhi.
4. Dengan adanya *spinning reserve* biaya operasi pada sistem akan semakin naik akibat pembatasan pembangkitan maksimum.

#### 5.2 Saran

Adapun saran untuk penelitian selanjutnya pada bidang operasi sistem tenaga berdasarkan hasil simulasi dan analisa pada tugas akhir ini, yaitu:

1. *Software* PowerGen dapat dikembangkan lebih lanjut untuk penambahan *transmission losses* pada perhitungan *multiobjective unit commitment*.
2. Adanya penambahan *ramp-rate* pada *software* perhitungan PowerGen.





*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN

Berikut merupakan implementasi algoritma *visual basic* untuk *multiobjective unit commitment* :

```

interface
{
{-----}
{      Unit Commitment Program from chapter 5      }
{      POWER GENERATION OPERATION AND CONTROL      }
{
{      Allen J. Wood and Bruce F. Wollenberg      }
{-----}
}
CONST
    INFINITY = 1.0E15;
    MAX_NUMUNITS = 11;
    MAX_NUMSTATES = 2048;
    MAX_NUMPERIODS = 24;

{
{-----}
{ <f> <s> Main Program }
PROCEDURE PROSES;
BEGIN

    assign(ff,'data.dum');
    rewrite(ff);

    DATADUMP(FF);
    FOR I := 1 TO NUMPER DO LOADSYS[I] :=
    PERIOD_DATA[I].SYSLOAD;
    IF CENTANG THEN BEGIN
        FOR J := 1 TO NUMUNT DO BEGIN
            UNTCOST[J] := (UNTEC3[J] + UNTEC2[J]*UNTMAX[J]*(1.0-
            UNTSPR[J]/100)
            +
            UNTMAX[J]*UNTMAX[J]*UNTEC1[J]*(1.0-
            UNTSPR[J]/100)*(1.0-UNTSPR[J]/100))*UNTFUELCOST[J];

            UNTEM[J] := (UNTEM3[J] + UNTEM2[J]*UNTMAX[J]*(1.0-
            UNTSPR[J]/100)

```

+ UNTMAX[J]\*UNTMAX[J]\*UNTEM1[J]\*(1.0-  
UNTSR[J]/100)\*(1.0-UNTSR[J]/100));

if cweight=3 then PRCPEN[J] :=UNTEM[J] / UNTCOST[J];  
if cweight=2 then PRCPEN[J] := UNTCOST[J] / UNTEM[J];  
if cweight=1 then PRCPEN[J] :=1;  
if cweight=0 then PRCPEN[J] :=1;  
END;

END

ELSE BEGIN

FOR J := 1 TO NUMUNT DO BEGIN

UNTCOST[J] := (UNTEC3[J] + UNTEC2[J]\*UNTMAX[J]\*(1.0-  
(SPR/100))  
+ UNTMAX[J]\*UNTMAX[J]\*UNTEC1[J]\*(1.0-(SPR/100))\*(1.0-  
(SPR/100)))\*UNTFUELCOST[J];

UNTEM[J] := (UNTEM3[J] + UNTEM2[J]\*UNTMAX[J]\*(1.0-  
(SPR/100))  
+ UNTMAX[J]\*UNTMAX[J]\*UNTEM1[J]\*(1.0-(SPR/100))\*(1.0-  
(SPR/100)));

if cweight=3 then PRCPEN[J] :=UNTEM[J] / UNTCOST[J];  
if cweight=2 then PRCPEN[J] := UNTCOST[J] / UNTEM[J];  
if cweight=1 then PRCPEN[J] :=1;  
if cweight=0 then PRCPEN[J] :=1;  
END;

END;

A:=PRCPEN;

SORT(A);

IF CENTANG THEN

FOR I := 1 TO NUMUNT DO UNTMAX[I] := UNTMAX[I]\*(1-  
UNTSR[J]/100)

ELSE

FOR I := 1 TO NUMUNT DO UNTMAX[I] := UNTMAX[I]\*(1-  
SPR/100);

FOR I := 1 TO NUMUNT DO

FOR J := 1 TO NUMUNT DO BEGIN

IF A[I] = PRCPEN[J] THEN BEGIN

NUNTMAX[I] := NUNTMAX[I-1] + UNTMAX [J];

END;



```

END;

STATE := [];
FOR I:=1 TO NUMUNT DO STATE := STATE + [I];
I := 0;
WHILE (STATE <> []) DO BEGIN
    I := I+1;
    UNTSTAT[ I ] := STATE;
    BORROW := TRUE;
    J := NUMUNT;
    WHILE BORROW DO
        IF (J IN STATE) THEN BEGIN
            STATE := STATE - [J];
            BORROW := FALSE
        END
    ELSE BEGIN
        STATE := STATE + [J];
        J := J - 1
    END;
    NUMSTAT := I;
END;
I:=1;
WHILE (STATE <> UNTSTAT[I]) AND (I < NUMSTAT) DO
I:=I+1;
IF (STATE = UNTSTAT[I]) THEN
    INITSTAT:=I
ELSE BEGIN
    NUMSTAT := NUMSTAT + 1;
    INITSTAT := NUMSTAT;
    WRITELN(FF, 'INITIAL STATE NOT IN PRIORITY LIST,
ADDING STATE TO LIST');
    UNTSTAT[ INITSTAT ] := STATE;
END;
IF (NMAX > NUMSTAT) THEN NMAX := NUMSTAT;
IF (XMAX > NUMSTAT) THEN XMAX := NUMSTAT;
WITH PERIOD_DATA[0] DO BEGIN
    FOR I := 1 TO NUMSTAT DO BEGIN
        FCOST[ I ] := INFINITY;
        FOR J := 1 TO NUMUNT DO UNTSTATNOW[ I, J ] := 0;
    END; { I }

```

```

FCOST[ INITSTAT ] := 0.0;
PCOST[ INITSTAT ] := 0.0;
PATH[ INITSTAT ] := 0;
UNTSTATNOW[ INITSTAT ] := UNTSTATINI;
END;
CONTINUE := TRUE;
K := 1;
WHILE (K <= NUMPER) AND CONTINUE DO WITH
PERIOD_DATA[K] DO BEGIN
    UNTSTATLAST := UNTSTATNOW;
    FOR I := 1 TO NUMSTAT DO BEGIN
        FCOST[ I ] := INFINITY;
        EMCOST[ I ] := INFINITY;
        PATH[ I ] := 0;
        FOR J := 1 TO NUMUNT DO UNTSTATNOW[ I, J ] := 0;
    END; { I }
    IF (K = 1) THEN BEGIN
        NLIM := 1;
        SETL[ 1 ] := INITSTAT;
    END
    ELSE
        BESTSTATES(SETL,PERIOD_DATA[K-1]
        ,FCOST,NLIM,NMAX);
    FOR J := 1 TO NUMUNT DO BEGIN
        IF LOADSYS[K]<NUNTMAX[J] THEN BEGIN
            if cweight = 3 then PR:=A[J];
            if cweight = 2 then PR:=A[J];
            if cweight =1 then PR:=1;
            if cweight =0 then PR:=1;
        BREAK;
        END;
    END;
    WRITELN(FF,' Price Penalty Factor = ',PR:4:4);
    EDC;
    writeln(ff,' ');
    BESTSTATES(SETX,PCOST,XLIM,XMAX);
    CONTINUE := FALSE;
    FOR X := 1 TO XLIM DO BEGIN
        XSTATE := SETX[ X ];
        FOR N := 1 TO NLIM DO BEGIN
            LSTATE := SETL[ N ];

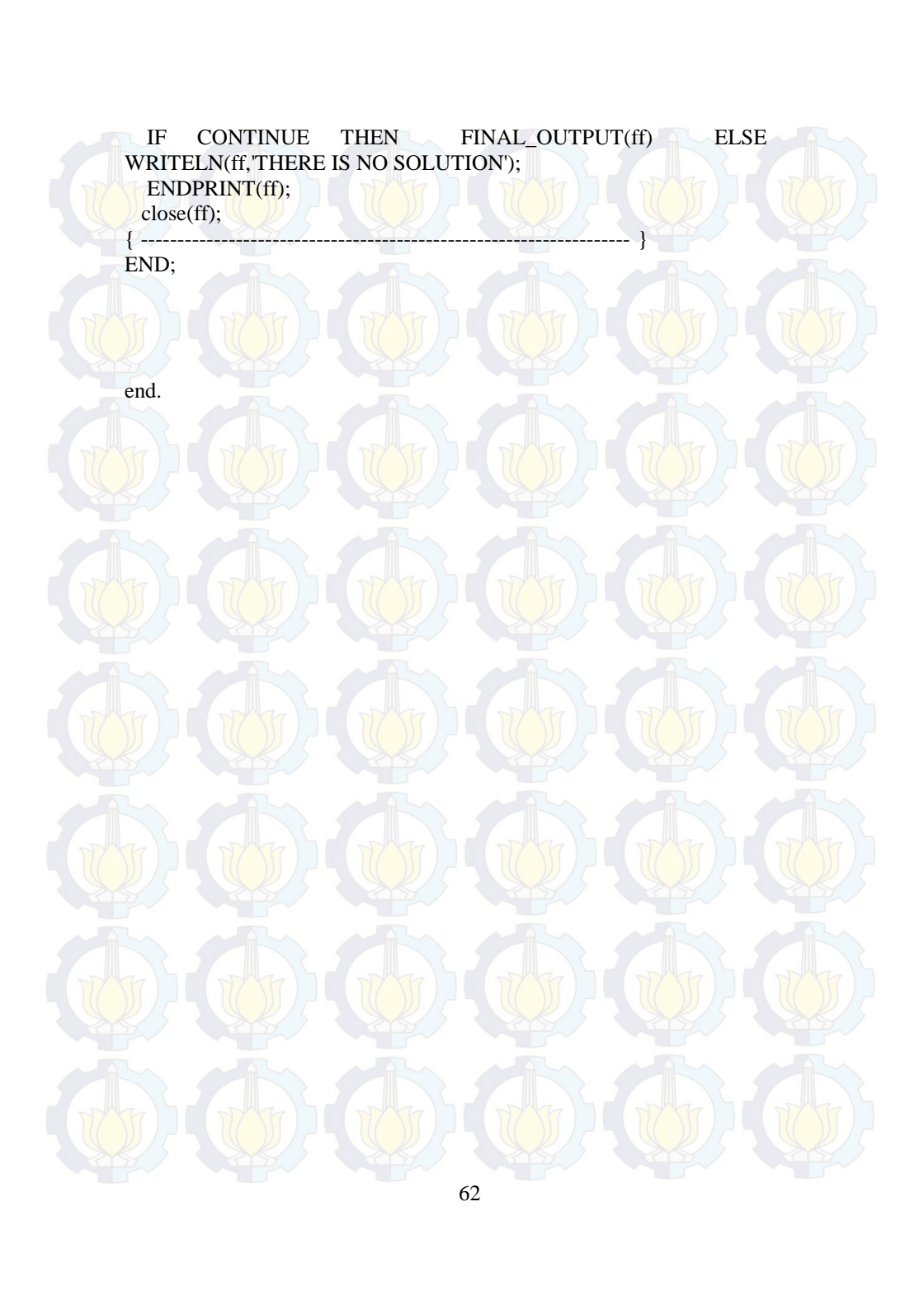
```

```

FEAS;
IF (VALID = 1) THEN BEGIN
    PATHCOST := PERIOD_DATA[ K-1].FCOST[ LSTATE ] +
SCOST +
        PCOST[ XSTATE ];
    PATHEM := PERIOD_DATA[ K-1].EMCOST[ LSTATE ]+
        ECOST[ XSTATE ];
    AVEPATH := (PATHCOST/PATHEM);
    AVECOST := (PCOST[ XSTATE ]/ECOST[ XSTATE ]);
    IF Cweight=1 THEN BEGIN
        IF (PATHEM <= EMCOST[ XSTATE ]) THEN BEGIN
            FCOST[ XSTATE ] := PATHCOST;
            EMCOST[ XSTATE ] := PATHEM;
            UNTSTATNOW[ XSTATE ] := UNTSTATTEMP;
            PATH[ XSTATE ] := LSTATE;
            CONTINUE := TRUE;
        END;
    END;
    IF CWEIGHT = 0 THEN BEGIN
        IF (PATHCOST <= FCOST[ XSTATE ]) THEN BEGIN
            FCOST[ XSTATE ] := PATHCOST;
            EMCOST[ XSTATE ] := PATHEM;
            UNTSTATNOW[ XSTATE ] := UNTSTATTEMP;
            PATH[ XSTATE ] := LSTATE;
            CONTINUE := TRUE;
        END;
    END;
    IF CWEIGHT = 2 THEN BEGIN
        IF AVEPATH<=AVECOST THEN BEGIN
            FCOST[ XSTATE ] := PATHCOST;
            EMCOST[ XSTATE ] := PATHEM;
            UNTSTATNOW[ XSTATE ] := UNTSTATTEMP;
            PATH[ XSTATE ] := LSTATE;
            CONTINUE := TRUE;
        END;
    END;
    END; { N }
    END; { X }
    K := K + 1;
    END; { K }

```





```
IF CONTINUE THEN FINAL_OUTPUT(ff) ELSE
WRITELN(ff,'THERE IS NO SOLUTION');
ENDPRINT(ff);
close(ff);
{-----}
END;
```

end.



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE141599**

***MULTIOBJECTIVE UNIT COMMITMENT OPTIMIZATION  
WITH EMISSION BY CONSIDERING SPINNING RESERVE  
USING LAGRANGIAN RELAXATION METHOD***

Mochammad Rizky Pratama  
NRP 2211100107

Supervisor  
Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.  
Dr. Rony Seto Wibowo, ST., MT.

ELECTRICAL ENGINEERING DEPARTEMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2015

# Optimasi *Multiobjective Unit Commitment* Dengan Emisi Mempertimbangkan *Spinning Reserve* Menggunakan Metode *Lagrangian Relaxation*

Mochammad Rizky Pratama  
2211100107

Dosen Pembimbing I : Prof. Ir. Ontoseno P, M.Sc., Ph.D.  
Dosen Pembimbing II : Dr. Rony Seto Wibowo, ST., MT.

## Abstrak:

Pembangkit termal mempunyai peranan penting dalam hal produksi tenaga listrik. Tetapi pembangkit termal menghasilkan dampak emisi yang mempengaruhi masalah lingkungan. *Multiobjective unit commitment* merupakan salah satu solusi selain *economic dispatch*, dengan fungsi biaya untuk menekan biaya dan juga terdapat *emission dispatch* dengan fungsi emisi untuk menekan jumlah emisi pada perencanaan operasi dalam sistem tenaga. Dalam pengoperasian pembangkit juga terdapat *security constraint* berupa *spinning reserve* untuk menjaga keandalan operasi sistem. *Multiobjective function* nantinya akan dimodelkan kedalam *single objective function* dengan menggunakan *price penalty factor*. Masalah *unit commitment* akan diselesaikan dengan metode *lagrangian relaxation*. Pada tugas akhir ini penyelesaian *multiobjective unit commitment* dikerjakan pada *Visual Basic* dan akan dibuat *interface* untuk menampilkan data masukan dan hasil akhir.

Kata Kunci : *Multiobjective Unit Commitment*, *Price Penalty Factor*, *Visual Basic*, *Spinning Reserve*, *Lagrangian Relaxation*.



# ***Optimasi Multiobjective Unit Commitment with Emission by considering Spinning Reserve using Lagrangian Relaxation Method***

M Rizky Pratama  
2211100107

Supervisor I : Prof. Ir. Ontoseno P, M.Sc., Ph.D.  
Supervisor II : Dr. Rony Seto Wibowo, ST., MT.

## **Abstract :**

*Thermal plant has a main role to bring a bright future in industrial world. And yet it has a side effect which is affecting environment. Multiobjective unit commitment is one of solutions to planning operation systems to maintain cost efficiency from cost function and emission yields from emission function. Both functions will be transformed into single objective function so that the function could be determining multiobjective unit commitment. To transform these functions, price penalty factor and weight should be included. Spinning reserve as security constraint will be involved to determine unit commitment. Unit commitment will be done with lagrangian relaxation method. And this final project uses a visual basic to create the interface and build algorithms.*

*Keywords : Multiobjective Unit Commitment, Price Penalty Factor, Visual Basic, Spinning Reserve, Lagrangian Relaxation.*

## KATA PENGANTAR

Pertama-tama marilah kita ucapkan puja dan puji syukur kepada Allah SWT karena limpahan rahmat yang telah diberikan oleh-Nya kepada kita semua terutama dalam penyelesaian Tugas Akhir ini dengan judul “Optimasi *Multiobjective Unit Commitment* dengan Emisi Mempertimbangkan *Spinning Reserve* Menggunakan Metode *Lagrangian Relaxation*”.

Dalam menyelesaikan Tugas Akhir ini penulis mendapatkan banyak sekali bantuan dan dukungan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang selalu meyakini penulis dalam kondisi apapun hingga dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Orang tua dan keluarga tercinta yang sangat pengertian dan mendukung penulis dalam berbagai bentuk seperti doa, moral, dan material.
3. Bapak Prof. Ir. Ontoseno Penangsang M.Sc., Ph.D., selaku dosen pembimbing 1 Tugas Akhir.
4. Bapak Dr. Rony Seto Wibowo ST., MT., selaku dosen pembimbing 2 Tugas Akhir.
5. Teman-teman seperjuangan yang saling membantu dalam penyelesaian Tugas Akhir ini.
6. Semua pihak yang tidak bisa disebutkan satu persatu yang turut membantu dalam penyelesaian Tugas Akhir ini baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa masih banyak kekurangan di dalam Tugas Akhir ini. Untuk itulah penulis menerima segala kritik dan saran yang berguna bagi penulis. Semoga buku Tugas Akhir ini berguna bagi orang lain. Sekian dan terima kasih.

Surabaya, Juni 2015

Penulis



*(Halaman ini sengaja dikosongkan)*



## TABLE OF CONTENT

<b>TITLE PAGE</b>	
<b>AUTHENTICATION</b>	
<b>APPROVAL SHEET</b>	
<b>ABSTRAK</b> .....	i
<b>ABSTRACT</b> .....	iii
<b>PREFACE</b> .....	v
<b>TABLE OF CONTENT</b> .....	vii
<b>ILLUSTRATIONS</b> .....	ix
<b>TABLES</b> .....	xi
<b>CHAPTER 1 INTRODUCTION</b> .....	1
1.1 Background .....	1
1.2 Research Aim .....	2
1.3 Problems .....	2
1.4 Problem Circumscription .....	2
1.5 Research Method .....	3
1.6 Process of Writing.....	4
1.7 Relevation .....	4
<b>CHAPTER 2 MULTIOBJECTIVE UNIT COMMITMENT WITH SPINNING RESERVE</b> .....	7
2.1 Generator .....	7
2.1.1 Generator Ability .....	7
2.1.2 Thermal Generator Characteristic .....	8
2.2 Load .....	9
2.2.1 Base Load.....	10
2.2.2 Medium Load.....	10
2.2.3 Peak Load.....	10
2.3 Unit Commitment .....	10
2.3.1 Minimum up-time .....	12
2.3.2 Minimum down-time .....	12
2.3.3 Spinning Reserve .....	12
2.3.4. Start-Up Cost .....	12
2.4 Unit Commitment Method.....	13
2.4.1 Priority List .....	13
2.4.2 Dynamic Programming .....	14
2.4.2 Lagrangian Relaxation .....	18
2.5 Economic Dispatch .....	20
2.6 Emission Dispatch .....	21

### **CHAPTER 3      MULTIOBJECTIVE UNIT COMMITMENT IMPLEMENTATION USING LAGRANGIAN RELAXATION METHOD .....**

3.1	Multiobjective Optimization.....	23
3.1.1	Price Penalty Factor.....	23
3.1.2	Weight .....	25
3.1.3	Multiobjective Algorithm .....	26
3.2	Lagrangian Relaxation Algorithm .....	26
3.3	Input Argument Multiobjective Unit Commitment .....	31
3.4	Syntax Software Multiobjective Unit Commitment using Delphi .....	32
3.5	Software utilizing to calculate Multiobjective Unit Commitment.....	33

### **CHAPTER 4 SIMULATION & RESULT .....37**

4.1	Case Study .....	38
4.1.1	Case 1 .....	38
4.1.2	Case 2 .....	40
4.1.3	Case 3 .....	42
4.1.4	Case 4 .....	44
4.2	Power Curve for Every Case. ....	47
4.3	Cases Analyst.....	49

### **CHAPTER 5 CONCLUSION & PROPOSITION.....53**

5.1	Conclusion .....	53
5.2	Proposition.....	53

### **REFERENCES .....55**

### **ENCLOSURE .....57**

### **BIOGRAPHY.....63**

## DAFTAR TABEL

<b>Tabel 3.1</b>	Argumen Input.....	31
<b>Tabel 3.2</b>	Sintaksis.....	32
<b>Tabel 4.1</b>	Data masukan <i>cost function</i> IEEE 39 bus.....	37
<b>Tabel 4.2</b>	Data masukan <i>emission function</i> IEEE 39 bus.....	37
<b>Tabel 4.3</b>	Data masukan minimum up- and down-time, start-up cost, dan inisial status.....	38
<b>Tabel 4.4</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BCC) Interval 1-8.....	39
<b>Tabel 4.5</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BCC) Interval 9-16.....	39
<b>Tabel 4.6</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BCC) Interval 17-24.....	40
<b>Tabel 4.7</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BEC) Interval 1-8.....	40
<b>Tabel 4.8</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BEC) Interval 9-16.....	41
<b>Tabel 4.9</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (BEC) Interval 17-24.....	42
<b>Tabel 4.10</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission) Interval 1-8.....	42
<b>Tabel 4.11</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission) Interval 9-16.....	43
<b>Tabel 4.12</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission) Interval 17-24.....	44
<b>Tabel 4.13</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission with SPR) Interval 1-8.....	45
<b>Tabel 4.14</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission with SPR) Interval 9-16.....	45
<b>Tabel 4.15</b>	Hasil <i>running</i> aplikasi <i>PowerGen</i> (Combined Economic Emission with SPR) Interval 17-24.....	46



## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Pemodelan pembangkit termal sederhana.....	7
<b>Gambar 2.2</b>	Kurva Karakteristik Input-Output Pembangkit Termal Uap.....	9
<b>Gambar 2.3</b>	Kurva Beban harian.....	9
<b>Gambar 2.4</b>	Pola sederhana <i>peak valley</i> beban.....	10
<b>Gambar 2.5</b>	Pola beban dengan aturan <i>unit commitment</i> .....	11
<b>Gambar 2.6</b>	Representasi <i>Dynamic Programming</i> .....	14
<b>Gambar 2.7</b>	Diagram alur <i>Dynamic Programming</i> .....	15
<b>Gambar 2.8</b>	Jalur pencarian DP.....	17
<b>Gambar 2.9</b>	Jalur Unit i untuk minimum <i>up- and down-time</i> .....	19
<b>Gambar 2.10</b>	Tampilan pengerjaan Delphi.....	22
<b>Gambar 3.1</b>	Diagram alur <i>price penalty factor</i> .....	24
<b>Gambar 3.2</b>	Diagram alur <i>lagrangian relaxation</i> .....	30
<b>Gambar 3.3</b>	<i>Main Display Multiobjective Unit Commitment</i> .....	33
<b>Gambar 3.4</b>	<i>Editor Input</i> .....	34
<b>Gambar 3.5</b>	<i>Input</i> untuk beban setiap interval.....	34
<b>Gambar 3.6</b>	<i>Weight Preferences Option</i> perhitungan pada <i>Software PowerGen</i> .....	34
<b>Gambar 3.7</b>	<i>Display</i> untuk <i>min up- dan down-time option</i> dan <i>method options</i> .....	35
<b>Gambar 4.1</b>	Kurva daya <i>Best Economic Commitment</i> .....	47
<b>Gambar 4.2</b>	Kurva daya <i>Best Emission Commitment</i> .....	47
<b>Gambar 4.3</b>	Kurva daya <i>Combined Economic Emission</i> .....	48
<b>Gambar 4.4</b>	Kurva daya <i>Combined Economic Emission with Spinning Reserve</i> .....	49
<b>Gambar 4.5</b>	Grafik biaya untuk kasus 1, 2 dan 3.....	49
<b>Gambar 4.6</b>	Grafik emisi untuk kasus 1, 2 dan 3.....	50

## BIOGRAFI



Penulis memiliki nama lengkap Mochammad Rizky Pratama. Penulis lahir di Samarinda tanggal 27 Juli 1994. Penulis adalah anak pertama dari pasangan Siswanto dan Tuti Alfiah. Setelah enam tahun menempuh pendidikan dasar di SDN 009 Sungai Kunjang, Penulis melanjutkan pendidikan ke SMPN 10 Samarinda pada tahun 2005. Mulai tahun 2008, penulis melanjutkan pendidikan di SMAN 3 Samarinda dan lulus pada tahun 2011. Pada tahun yang sama, penulis masuk ke Jurusan S1 Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS), Surabaya. Di semester kelima, penulis mulai mendalami Bidang Studi Sistem Tenaga. Penulis juga aktif dalam berbagai kegiatan diantaranya UKM Robotika ITS, UKM Maritime Challenge, dan Workshop Teknik Elektro ITS. Penulis dapat dihubungi melalui alamat email: [rizkypratama.elits@gmail.com](mailto:rizkypratama.elits@gmail.com)

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Energi listrik telah menjadi kebutuhan primer bagi kehidupan manusia saat ini, sehingga menyebabkan permintaan energi listrik terus meningkat setiap tahunnya. Semakin berkembangnya teknologi menjadi salah satu alasan peningkatan permintaan daya listrik ini. Selain itu terus bertambahnya jumlah penduduk turut menghadirkan masalah dalam kenaikan beban. Hal tersebut menjadi tantangan produsen untuk mengelola dan mengatur pembangkitan agar *supply* dan *demand* terpenuhi, sehingga produsen listrik tidak mengalami kerugian.

Peningkatan kebutuhan tenaga listrik harus diimbangi dengan pengelolaan sumber daya yang baik. Sehingga mampu menyediakan tenaga listrik ekonomis dan tetap menjaga kontinuitas tenaga listrik. Penjadwalan pembangkit atau *unit commitment* merupakan metode yang digunakan untuk menekan biaya yang dibutuhkan untuk membangkitkan daya listrik. Faktor biaya, pada pembangkit termal mampu mempengaruhi hasil *unit commitment*. Pada setiap persoalan *unit commitment*, setiap unit pembangkit mempunyai karakteristik yang berbeda-beda baik dari fungsi biaya, minimum *up-* dan *down-time* dan biaya *start-up* pembangkit. Metode yang digunakan pada tugas akhir ini adalah *lagrangian relaxation*. Pada metode ini selain penentuan unit mana saja yang nyala dan tidak nyala, metode ini juga mengoptimalkan daya pembangkitan setiap unit yang nyala.

Selain dari biaya operasi, faktor lain yang penting untuk dianalisis adalah faktor emisi yang berdampak pada lingkungan. Tugas akhir ini menggunakan pembangkit termal untuk membangkitkan daya listrik yang notabene menggunakan bahan bakar fosil sebagai bahan bakar utama. Pembangkit termal tersebut dapat mencemarkan lingkungan dengan emisi yang dihasilkan terutama bagi atmosfer. Maka dari itu penting untuk membangkitkan daya listrik pada pembangkit termal dengan memperhatikan tidak hanya pada segi biaya tetapi dari segi lingkungan juga. Beberapa metode sudah dilakukan oleh para peneliti untuk menyelesaikan permasalahan tersebut. Seperti melakukan teknik probabilitas untuk mendapatkan penyelesaian optimal pada satu masalah dengan dua objek. *Multiobjective* merupakan cara lain penyelesaian untuk masalah lebih dari satu fungsi objektif. Dengan adanya dua fungsi yang dioptimalkan, yaitu fungsi biaya dan fungsi emisi, kedua fungsi tersebut akan dibuat menjadi *single objective function*.



Perhitungan *multiobjective unit commitment* membutuhkan proses yang panjang oleh karena itu pada tugas akhir ini diusulkan menggunakan Delphi. Delphi merupakan bahasa pemrograman berbasis Windows yang menyediakan fasilitas pembuatan aplikasi visual. Delphi memberikan kemudahan dalam menggunakan kode program, kompilasi yang cepat, penggunaan data unit ganda untuk pemrograman modular, pengembangan perangkat lunak, pola desain yang menarik serta diperkuat dengan bahasa pemrograman yang terstruktur dalam bahasa pemrograman Object Pascal. Delphi memiliki tampilan khusus yang didukung suatu lingkup kerja komponen Delphi untuk membangun suatu aplikasi dengan menggunakan *Visual Component Library* (VCL). Sebagian besar pengembangan Delphi menuliskan dan mengkompilasi kode program dalam IDE (*Integrated Development Environment*). Sehingga pada tugas akhir ini akan menghasilkan sebuah aplikasi perhitungan yang menggunakan bahasa yang dapat dipahami oleh pengguna. Aplikasi penyelesaian *multiobjective unit commitment* dengan memperhatikan *spinning reserve* yang lebih mudah untuk berinteraksi dengan pengguna.

## 1.2 Tujuan Penelitian

Tujuan yang ingin dicapai pada Tugas Akhir ini adalah sebagai berikut:

1. Membuat aplikasi perhitungan *multiobjective unit commitment* dengan menggunakan metode *lagrangian relaxation* berbasis DELPHI.
2. Melihat pengaruh emisi dan *spinning reserve* pada perhitungan MUC

## 1.3 Permasalahan

Dalam Tugas Akhir ini diharapkan permasalahan aplikasi perhitungan *multiobjective unit commitment* yang dapat dengan mudah berinteraksi dengan pengguna dapat berjalan dengan memperhatikan *spinning reserve* yang diselesaikan dengan metode *lagrangian relaxation*.

## 1.4 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Aplikasi perhitungan dikerjakan dengan menggunakan DELPHI
2. Metode yang digunakan untuk menyelesaikan permasalahan *multiobjective unit commitment* adalah metode *lagrangian relaxation*
3. Fungsi yang diperhitungkan adalah fungsi biaya dan fungsi emisi

4. Biaya *start-up cost* merupakan *fixed cost*, tidak memperhatikan berapa lama unit tersebut telah di *shut down*.
5. Tidak memperhatikan *transmission losses*

## 1.5 Metode Penelitian

Pada tugas akhir ini akan dilakukan penelitian tentang *multiobjective unit commitment* dengan memperhatikan batasan *spinning reserve* untuk rentang waktu tertentu. *Lagrangian relaxation* digunakan untuk menentukan hasil *unit commitment*.

### 1. Studi Literatur

Literatur - literatur yang digunakan berasal dari jurnal, prosiding dan *text book*. Dalam studi literatur dipelajari teori pendukung yang berisi tentang *unit commitment* yang melihat pengaruh dari fungsi biaya, fungsi emisi dan *spinning reserve*, metode *lagrangian relaxation* yang dapat diterapkan dalam penyelesaian output pembangkitan dalam persamaan *multiobjective unit commitment* dan pembuatan *interface* menggunakan Delphi.

### 2. Pengumpulan Data dan Perhitungan

Data yang bersangkutan dalam permasalahan *unit commitment* seperti batasan maximum pembangkit, batasan minimum pembangkit, koefisien fungsi bahan bakar, koefisien fungsi emisi, data beban tiap periode dalam interval waktu 1 jam, *fuelcost*, minimum *up-time*, minimum *down-time*, dan *spinning reserve* serta bobot. Berdasarkan formulasi yang telah diketahui dilakukan perhitungan terlebih dahulu untuk menunjang pemodelan dan simulasi.

### 3. Perencanaan dalam Pembuatan Program

Dari data perhitungan tersebut dibuatlah struktur perancangan logika berupa diagram alir untuk proses pembuatan program yang nantinya akan diimplementasikan kedalam bahasa pemrograman Delphi.

### 4. Simulasi Analisa Data

Setelah pembuatan program telah selesai dan menjadi sebuah aplikasi perhitungan, simulasi akan dicoba dengan menggunakan data inputan yang telah tersedia. Simulasi bertujuan untuk melihat aplikasi perhitungan *multiobjective unit commitment* yang dibuat telah sesuai dengan teori perhitungan yang ada. Dari perjalanan simulasi akan didapatkan data yang akan dianalisa kebenarannya. Analisa bertujuan untuk memastikan hasil perhitungan program telah sesuai dengan metode yang digunakan. Untuk mendapatkan hasil perhitungan yang sesuai, hasil

dari data outputan akan dibandingkan dengan data *output* referensi dan batasan yang telah ditentukan.

## 5. Penulisan Buku

Hasil penelitian yang telah dilakukan akan dilaporkan dalam sebuah buku laporan Tugas Akhir. Isi dari laporan berdasarkan kesimpulan dari analisis yang telah didapat beserta tahapan yang ada di dalamnya.

### 1.6 Sistematika Penulisan

#### BAB 1 PENDAHULUAN

Bab ini membahas Latar Belakang, Tujuan Penelitian, Permasalahan, Batasan Masalah, Metode Penelitian, Sistematika Penulisan, dan Relevansi pada laporan Tugas Akhir.

#### BAB 2 MULTIOBJECTIVE UNIT COMMITMENT

Bab ini berisikan teori penunjang dalam Tugas Akhir. Teori yang dibahas meliputi Generator, Beban, Karakteristik Pembangkit, *unit commitment*, *spinning reserve*, *dynamic programming*, *lagrangian relaxation*, *economic dispatch*, *emission dispatch*, dan bahasa pemrograman VB/Delphi

#### BAB 3 IMPLEMENTASI MULTIOBJECTIVE UNIT COMMITMENT DENGAN METODE LAGRANGIAN RELAXATION

Bab ini berisikan penjelasan mengenai tentang *multiobjective optimization* dengan *price penalty factor* serta penggunaan *dynamic programming* dan *lagrangian relaxation* yang diterapkan pada permasalahan *multiobjective unit commitment*.

#### BAB 4 SIMULASI DAN ANALISIS

Bab ini berisi data hasil *running* aplikasi MUC yang hasil keluaran data dianalisis dengan cara membandingkan hasil *output* setiap kasus.

#### BAB 5 PENUTUP


Bab terakhir ini memberikan kesimpulan dari simulasi dan analisis yang telah dilakukan dalam Tugas Akhir.

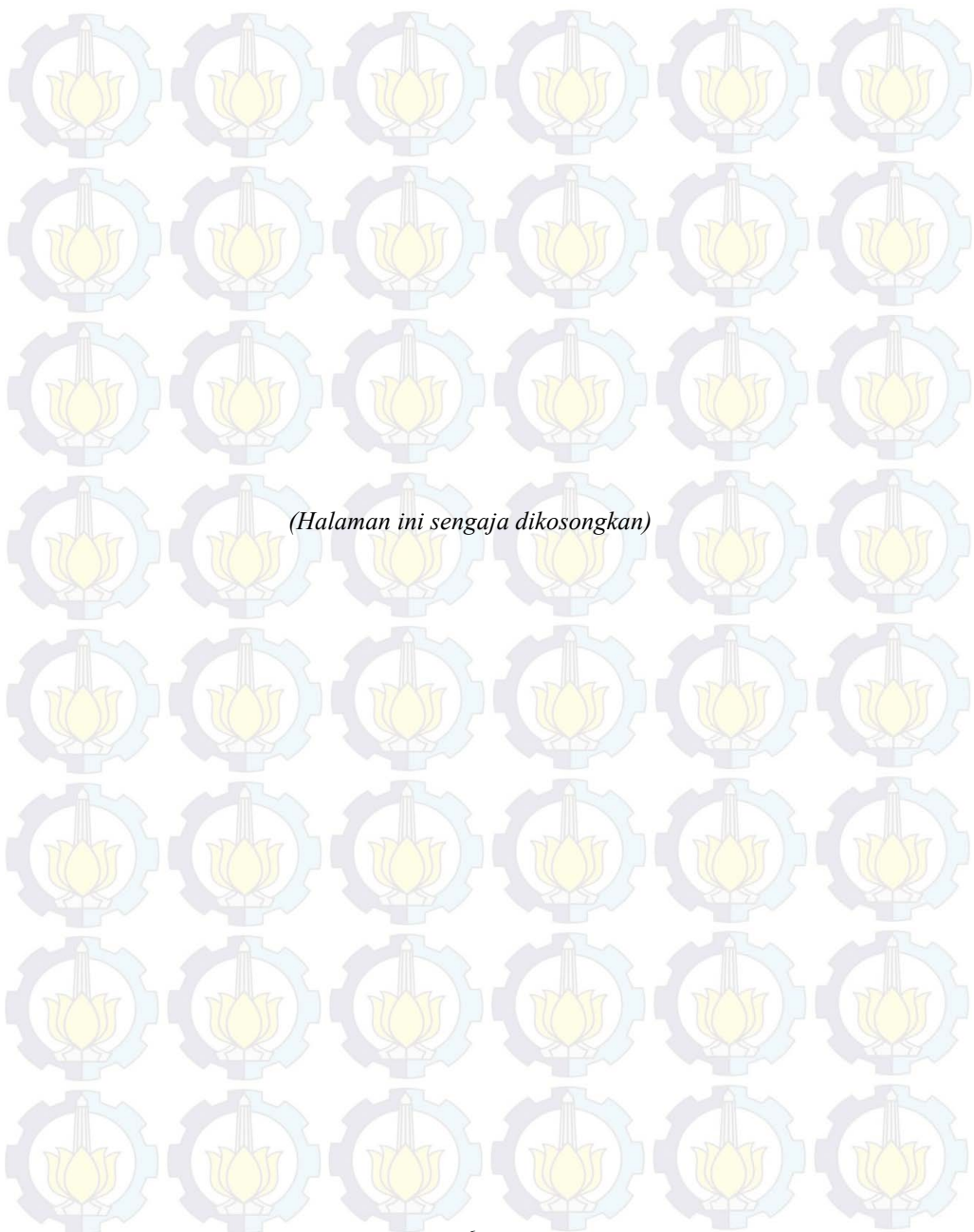
### 1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat memberikan kontribusi sebagai berikut:

1. Dapat memberikan manfaat dalam kemudahan melakukan perhitungan optimalisasi *multiobjective unit commitment* menggunakan *lagrangian relaxation* dengan memperhatikan *spinning reserve*.
2. Dapat menambah penguasaan ilmu dan teknologi di bidang optimalisasi pembangkitan sistem tenaga listrik.



- 
3. Dapat menjadi referensi Tugas Akhir untuk mahasiswa yang akan mengambil Tugas Akhir untuk mengembangkan aplikasi perhitungan *multiobjective unit commitment* dengan permasalahan yang berbeda.

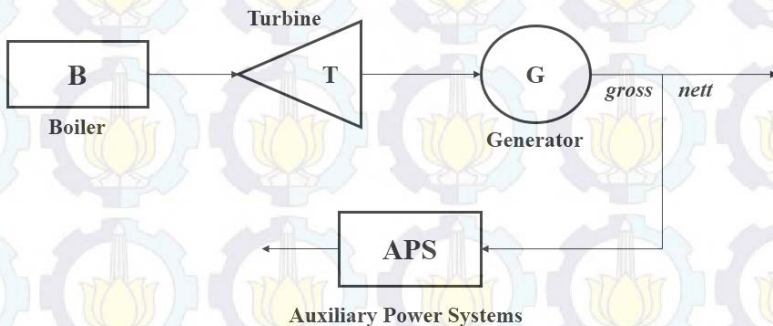


## BAB 2

### **MULTIOBJECTIVE UNIT COMMITMENT DENGAN SPINNING RESERVE**

#### **2.1 Generator**

Di sistem tenaga listrik generator merupakan salah satu komponen penting untuk mengoperasikan sistem tenaga listrik. Generator merupakan pembangkit listrik yang menghasilkan output daya untuk disalurkan ke beban. Generator yang paling umum digunakan ialah generator 3 phasa atau generator sinkron. Dalam sebuah unit pembangkitan, beberapa generator dioperasikan secara paralel dan dihubungkan ke bus jaringan untuk memenuhi *demand load*. Dari semua elemen dalam sistem tenaga listrik, biaya bahan bakar pada generator merupakan elemen terbesar dari semua jenis biaya yang diperlukan dalam sebuah usaha pembangkitan tenaga listrik. Sehingga pengoptimalan *scheduling* di semua unit pembangkit pada sistem tenaga listrik sangat diperlukan untuk menekan biaya bahan bakar agar mendapatkan nilai dari fungsi biaya pembangkitan yang minimum.



**Gambar 2.1** Pemodelan pembangkit termal sederhana

##### **2.1.1 Batasan kemampuan generator**

Kemampuan generator untuk membangkitkan daya listrik mempunyai aturan yang harus diperhatikan.  $P_{min}$  dan  $P_{max}$  pada generator merupakan aturan yang harus dipenuhi untuk syarat membangkitkan daya listrik. Generator harus membangkitkan daya dalam rentang  $P_{min}$  dan  $P_{max}$  agar pengoperasian generator lebih ekonomis, efisien dan tidak menimbulkan kerusakan pada generator. Generator tidak boleh dioperasikan dibawah  $P_{min}$  (*under operation*) dan diatas  $P_{max}$  (*over operation*) seperti pada persamaan(2.0).



$$P_{i_{min}} \leq P_{gen_i} \leq P_{max_i} \quad (2.0)$$

Keterangan :

$P_{i_{min}}$  : Daya minimum unit i

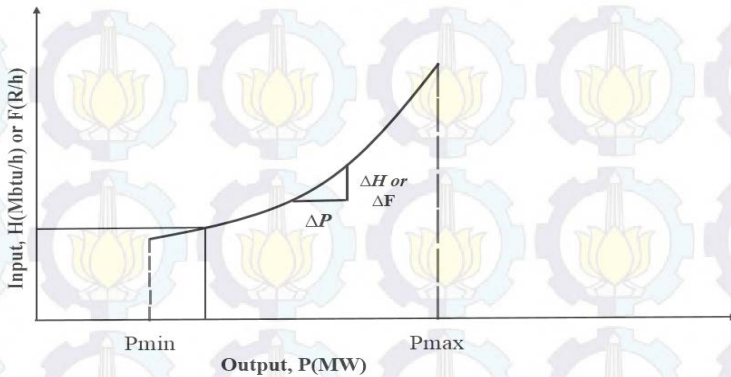
$P_{max_i}$  : Daya maksimum unit i

$P_{gen_i}$  : Daya yang dibangkitkan unit i

### 2.1.2 Karakteristik Pembangkit Termal

Hal dasar yang perlu diperhatikan untuk mendapatkan solusi ekonomis dari pembangkit termal adalah menghitung karakteristik *input-output* pada pembangkit termal. Untuk menghitung biaya pengeluaran pada pembangkit yang beroperasi, ada cukup banyak parameter yang perlu diperhatikan. Karakteristik input-output suatu pembangkit tidak hanya mencakup pada sistem tenaga listrik, tetapi terdapat juga elemen pendukung yang mempengaruhi karakteristik input-output. Biaya *maintenance* dan biaya operasi sudah termasuk dalam karakteristik input-output biaya operasi, *R per hour*, setiap unit. Selain itu juga terdapat biaya tambahan yang dimasukkan ke dalam *fixed cost*.

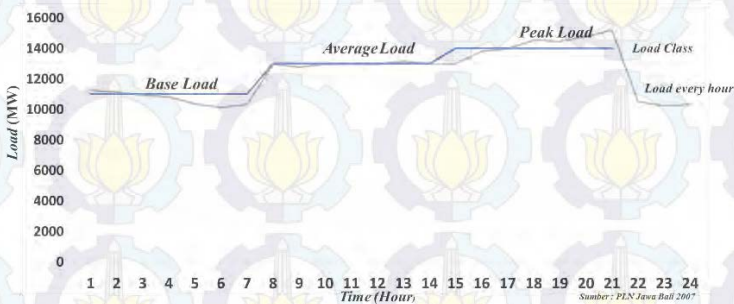
Gambar 2.2 menunjukkan karakteristik input-output dalam bentuk ideal berupa kurva *smooth convex*. Pada elemen input unit di Gambar 2.2 ditunjukkan bahwa input dapat dijadikan sebagai jumlah energi panas yang dibutuhkan per jam atau sebagai total biaya per jam. Dan pada output berupa besar daya yang dihasilkan tiap unit. Data karakteristik input-output didapat dari perhitungan-perhitungan sistem pembangkit atau dari uji tingkat panas. Pada Gambar 2.3, kurva menunjukkan hubungan perubahan daya pembangkitan yang dihasilkan dengan konsumsi bahan bakar yang dibutuhkan. *Incremental rate* =  $\Delta H / \Delta P$  atau *incremental rate* =  $\Delta F / \Delta P$ . Incremental rate dinyatakan dengan satuan Btu/kWh( $\Delta H / \Delta P$ ) atau R/kWh( $\Delta F / \Delta P$ ).



**Gambar 2.2** Kurva Karakteristik Input-Output Pembangkit Termal Uap

## 2.2 Beban

Salah satu yang menjadi tolak ukur kualitas sistem tenaga listrik dikatakan baik apabila pada sisi beban dikatakan terlayani dengan baik. Beban merupakan tujuan akhir dari daya listrik yang dibangkitkan oleh generator. Menurut pola konsumsi beban umumnya dibagi menjadi beberapa jenis yaitu, beban industri, beban perumahan dan beban komersil. Pada beban komersil dan beban perumahan biasanya terdiri dari sistem penerangan dan pendinginan dalam area yang luas. Beban ini tidak bergantung pada frekuensi karena mengonsumsi daya reaktif yang sangat kecil. Pola pembebanan pembangkit umumnya didasarkan pada fleksibilitas operasi, level biaya dan kapasitas pembangkitan karakteristik pembangkit. Menurut level, beban dikategorikan sebanyak tiga kategori antara lain:



**Gambar 2.3** Kurva Beban harian

### 2.2.1 Beban Dasar

Pada beban dasar, pembangkit yang dinyalakan pada sistem ialah merupakan kategori pembangkit yang memiliki daya keluaran besar, biaya kapital tinggi dan biaya operasi rendah. Pembangkit *base load* adalah pembangkit dengan 5000 jam operasi rata-rata per tahun dengan *capacity factor* lebih dari 57%.

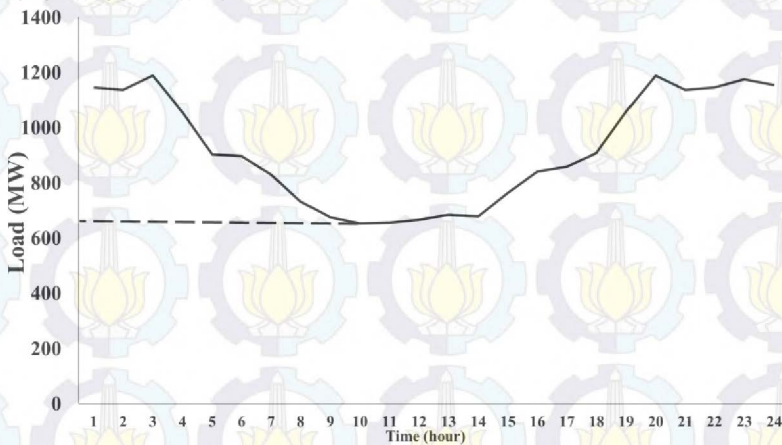
### 2.2.2 Beban Menengah

Pada level beban menengah, pembangkit yang beroperasi umumnya merupakan pembangkit *combined cycle*, pembangkit berbahan bakar gas dan pembangkit tua yang efisiensinya sudah menurun. Pembangkit *mid-range load* rata-rata beroperasi selama 2000 jam per tahun dengan  $23\% < \text{capacity factor} < 57\%$ .

### 2.2.3 Beban Puncak

Beban puncak ialah nilai beban tertinggi pada sistem jaringan tenaga listrik. Pada beban puncak pembangkit *peak load* dioperasikan untuk memenuhi beban pada waktu beban maksimum (beban puncak). Periode beban puncak tidak selalu sama. Pembangkit yang beroperasi pada beban puncak merupakan pembangkit dengan karakteristik biaya operasi yang tinggi dan biaya kapital rendah. Pengoperasian pembangkit *peak load* rata-rata beroperasi selama kurang dari 2000 jam. Pembangkit *peak load* menggunakan pembangkit listrik berbahan minyak, air dan *pump storage*.

## 2.3 Unit Commitment

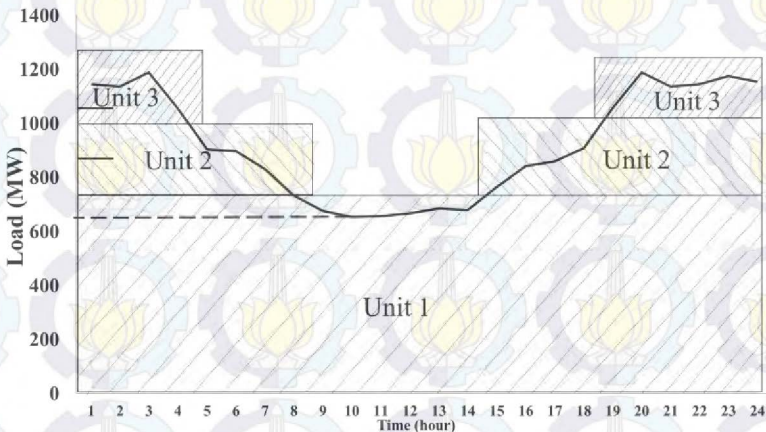


Gambar 2.4 Pola sederhana *Peak valley* beban



Siklus kerja manusia untuk melakukan kegiatan berdampak langsung kepada penggunaan energi listrik. Sehingga hal ini menimbulkan siklus harian, bulanan bahkan tahunan. Kasus umum yang terjadi pada siklus harian sistem tenaga listrik, pada siang hari hingga menjelang malam umumnya jumlah permintaan listrik bernilai rendah bila dibandingkan dengan konsumsi daya listrik pada malam hari, dikarenakan siklus industri yang bekerja pada waktu siang hari. Dan pada akhir minggu jumlah penggunaan listrik cenderung berkurang dibandingkan dengan hari-hari kerja.

Akibat dari siklus penggunaan listrik maka diperlukan adanya *unit commitment*. *Unit commitment* merupakan susunan penjadwalan setiap unit pembangkit yang terhubung satu sama lain untuk setiap periode dikarenakan permintaan beban yang berubah-ubah. *Unit commitment* merupakan formulasi dari kombinasi dua sub-masalah yaitu menentukan unit-unit yang aktif dan tidak aktif untuk setiap interval dan jumlah daya yang dibangkitkan setiap unit yang aktif. Dengan adanya *unit commitment* jumlah *cost* yang dapat dihemat cukup besar. Gambar 2.5 menunjukkan penjadwalan unit yang sederhana dengan melihat *demand load*. ditunjukkan bahwa unit 1 merupakan penopang untuk beban dasar. Unit 2 dinyalakan ketika unit 1 sudah tidak mampu menopang *average load* sehingga unit 2 dinyalakan dan untuk beban puncak, semua unit dinyalakan untuk membangkitkan daya listrik yang dibutuhkan untuk memenuhi *demand load*.



**Gambar 2.5** Pola beban dengan aturan *unit commitment*

Tetapi *unit commitment* mempunyai *constraint* yang harus diperhatikan antara lain adalah :

### 2.3.1 Minimum up-time

Setelah unit beroperasi, unit tersebut tidak dapat dimatikan secepatnya, hal ini dikarenakan perubahan temperatur yang perlahan sehingga menghalangi *decommit* unit pada beberapa waktu interval.

### 2.3.2 Minimum down-time

Unit tidak dapat dinyalakan seketika setelah unit *decommit*, adanya minimum waktu membuat unit bisa di nyalakan lagi setelah memenuhi interval waktu tersebut.

### 2.3.3 Spinning Reserve

Keandalan sebuah sistem tenaga listrik diukur dari jumlah beban yang diminta dan jumlah daya yang dikirimkan, apabila selisih *demand load* dan daya yang ditransmisikan berjumlah mendekati nol maka sistem dalam keadaan yang baik. Tetapi apabila pada sistem terdapat gangguan pada satu unit yang mengakibatkan unit tersebut tidak bekerja, maka dapat mengakibatkan *drop* frekuensi yang sangat besar. *Spinning reserve* dideskripsikan sebagai jumlah semua pembangkitan unit yang mungkin dapat dilakukan dikurangi dengan beban saat ini dan *transmission loss*. Pada sistem, *spinning reserve* harus ditempatkan pada lokasi yang menuntut respon yang cepat ketika terdapat satu unit yang hilang.

$$P_{i \min}^t \leq P_{i \text{ generate}}^t + SPR_i \leq P_{i \max}^t \quad (2.2)$$

Keterangan :

- $P_{i \min}^t$  : Daya minimum yang dapat dibangkitkan unit i saat t  
 $P_{i \max}^t$  : Daya maksimum yang dapat dibangkitkan unit i saat t  
 $P_{i \text{ generate}}^t$  : Daya yang dibangkitkan unit i saat t  
 $SPR_i$  : *Spinning reserve* unit i

### 2.3.4. Start-Up Cost

Biaya *start-up* pada pembangkit merupakan besarnya biaya yang diperlukan untuk menyalakan unit pembangkit saat interval sebelumnya sedang *decommit* dan saat interval selanjutnya unit harus *commit*. Jumlah *start-up cost* bisa bervariasi dari biaya maksimum *cold start* hingga biaya yang lebih kecil jika unit sudah *decommit* dengan interval yang kecil. Ada dua tindakan pada unit termal saat *decommit*. Pertama unit dibiarkan

dingin untuk beberapa interval dan dipanaskan kembali untuk beroperasi saat *commit*. Kedua unit diberikan energi yang cukup untuk menjaga temperatur agar biaya *start-up* unit murah.

## 2.4 Metode *Unit Commitment* [1]

Dalam penyelesaian *unit commitment* terdapat banyak metode yang bisa digunakan untuk mendapatkan hasil *unit commitment*, *economic dispatch*, *emission dispatch* dan *combined economic emission dispatch*.

### 2.4.1 *Priority List*

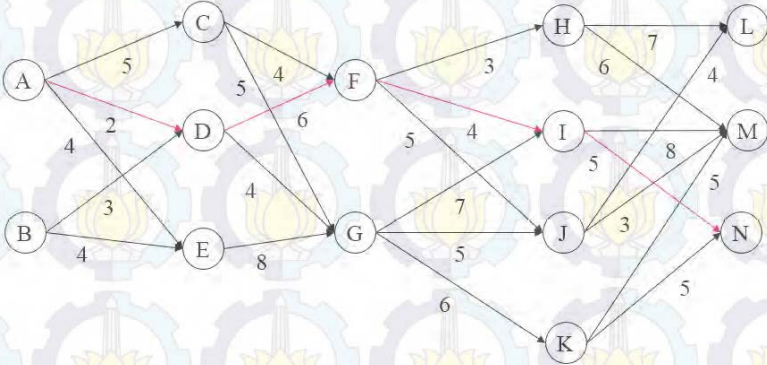
Metode solusi penyelesaian *unit commitment* yang paling sederhana adalah pembuatan *priority list* untuk unit-unit pembangkit pada sistem. Skema *priority list* dibuat dengan algoritma yang sangat sederhana seperti berikut :

- Pada setiap periode ketika beban turun, harus ditentukan untuk memilih unit, A, yang harus di *shut down*, dengan memperhatikan *spinning reserve*, pada *priority list* agar nilai *supply* dan *demand* seimbang. Jika tidak maka operasi berjalan seperti sebelumnya.
- Tentukan jumlah periode, H, sebelum unit A yang mati dinyalakan kembali.
- Jika H bernilai kurang dari minimum *down time* unit A saat unit A dibutuhkan kembali maka unit tetap dalam keadaan “on” saat terakhir kali unit A dimatikan.
- Hitung kedua biaya yang didapat, pertama jumlah biaya produksi untuk periode H selanjutnya dengan unit A yang menyala. Lalu hitung kembali jumlah biaya produksi untuk periode H selanjutnya dengan unit A yang dimatikan dengan tambahan *start-up cost*. Bandingkan kedua biaya produksi tersebut dan pilih yang paling murah.

Skema *priority list* dapat ditingkatkan dengan berbagai cara salah satunya membuat group-group yang berisilebih dari satu unit dengan berbagai *constraint* yang ada.



### 1.4.2 Dynamic Programming



**Gambar 2.6** Representasi *Dynamic Programming*

Pada *dynamic programming* untuk *unit commitment*, pendekatan yang dilakukan adalah :

1. *State* terdiri dari susunan beberapa unit dengan spesifikasi unit yang beroperasi dan tidak beroperasi.
2. Biaya *start-up* tidak bergantung pada berapa lama waktu unit tersebut telah di *shut-down*.
3. Tidak ada biaya untuk melakukan *unit decommitment*.
4. Setiap interval *state* harus memenuhi spesifikasi minimum kapasitas jumlah untuk unit yang dioperasikan.

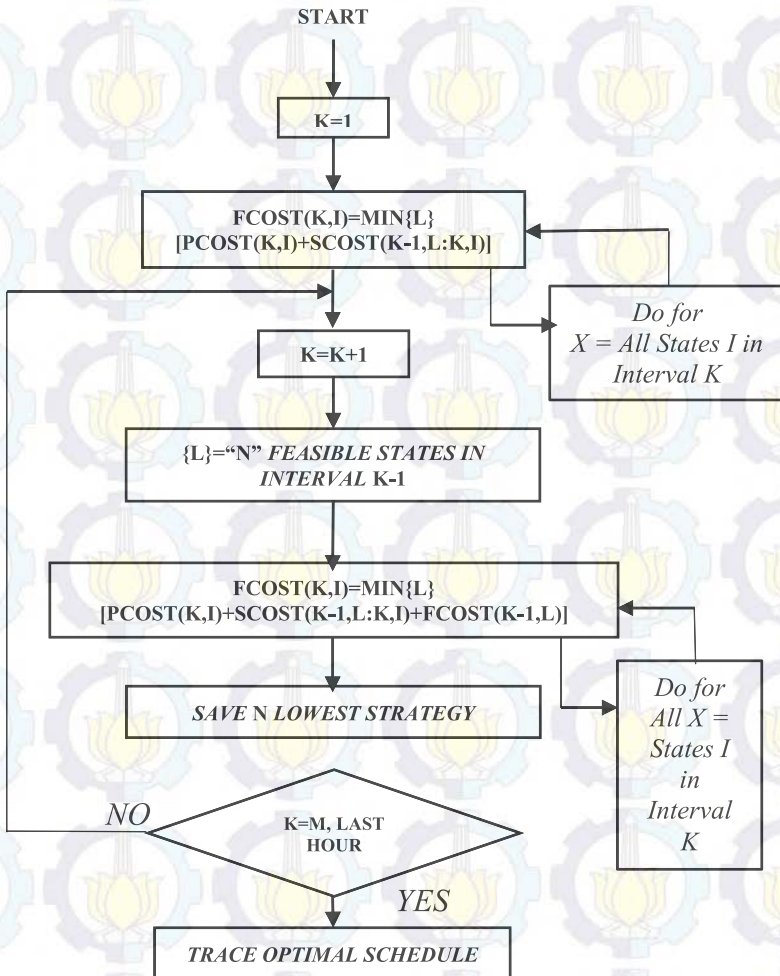
Sebuah solusi dari *dynamic programming* adalah kumpulan dari *state* pada interval pertama hingga interval terakhir. Dan *state* yang menjadi solusi dari permasalahan merupakan *feasible state*. Dan jumlah maksimum *state* yang dapat dimasukkan ke dalam penyelesaian *dynamic programming* secara matematis adalah sebagai berikut:

$$\text{Maks.State} = 2^n - 1 \quad (2.3)$$

Dimana  $n$  merupakan jumlah unit yang terhubung ke sistem dan bisa beroperasi.

*Dynamic programming* yang digunakan pada *unit commitment* adalah *forward dynamic programming*. Pada *forward dynamic programming* pendekatan yang dilakukan lebih menguntungkan apabila dibandingkan dengan *backward dynamic programming*. Pendekatan pada *forward dynamic programming* dilakukan dengan pengecekan hasil pada

state sebelumnya dimulai dari waktu inisial hingga akhir interval. Alasan praktis lain untuk menggunakan *forward dynamic programming* adalah mudah untuk menspesifikasikan constraint dan mengkomputasi perhitungan dimulai pada kondisi inisial.



**Gambar 2.7** Diagram alur *dynamic programming* [1]

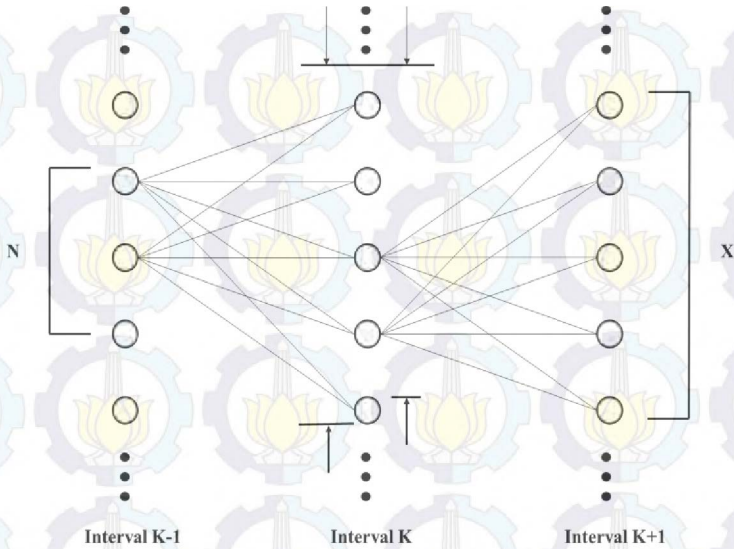
Pada Gambar 2.7 state (K,I) merupakan kombinasi unit pembangkit ke-I saat interval K. Untuk pendekatan *forward dynamic programming* kita mendefinisikan strategi sebagai transisi atau jalur, dari state-state sebelumnya ke state interval selanjutnya. Ada dua variabel pada *forward dynamic programming*, yang pertama adalah X. X merupakan jumlah kombinasi unit yang dicari pada interval tersebut. Yang kedua adalah N, merupakan strategi atau jumlah kombinasi unit yang disimpan pada setiap step untuk menentukan apakah state sebelumnya merupakan penyelesaian yang optimal untuk interval selanjutnya.

Jumlah nilai X dan N maksimal yang dapat dilakukan dalam penyelesaian juga sebesar jumlah maksimal state pada permasalahan. Akan tetapi jumlah kombinasi yang dapat dilakukan setiap interval harus merupakan state yang *feasible* untuk konstrain yang berlaku. Gambar 3.3 merupakan contoh sederhana untuk menggunakan *forward dynamic programming*. Pada Gambar 2.8, jumlah state, X, yang digunakan untuk solusi optimal berjumlah 5 dan jumlah kombinasi unit yang disimpan, N, untuk mendapatkan hasil atau jalur yang optimal beserta konstrain yang terpenuhi berjumlah 3.

*Dynamic programming* tidak hanya berperan untuk menentukan jumlah state yang akan dioptimalkan tetapi juga mengatasi *constraint* yang ada pada *unit commitment* seperti minimum *up-* dan *down-time*. Setelah semua state yang *feasible* sudah dioptimalkan maka *dynamic programming* akan mencari hasil fungsi yang paling optimal dengan cara menentukan jalur strategi mulai dari inisial interval hingga akhir interval.

Pada *unit commitment*, pendekatan *dynamic programming* yang dilakukan ialah *forward DP*. Hal ini dikarenakan pada *unit commitment* terdapat *start-up cost* yang menambah total biaya apabila terdapat unit yang harus dinyalakan.





**Gambar 2.8** Jalur pencarian DP [1]

Alasan lain *forward DP* digunakan ialah pada *unit commitment* terdapat inisial status per unit yang mengharuskan perhitungan dimulai dari awal agar memudahkan perhitungan.

$$F_{cost}(K, I) = \min[P_{cost}(K, I) + S_{cost}(K - 1, L; K, 1) + F_{cost}(K - 1, L)] \quad (2.4)$$

Keterangan:

- |                            |  |
|----------------------------|--|
| K                          | : Periode atau jam ke K  |
| I                          | : Kombinasi semua unit <i>state</i> I                                      |
| $F_{cost}(K, I)$           | : Total biaya optimum yang diperlukan untuk sampai ke state I pada stage K |
| $P_{cost}(K, I)$           | : Biaya produksi pada state I untuk stage K                                |
| $S_{cost}(K - 1, L; K, I)$ | : Transisi biaya dari state (K - 1), stage L ke state K, stage I           |

Gambar 2.4 menunjukkan cara kerja *forward dynamic programming*. Pada Gambar 2.4, A,B,C,D,...,N merupakan state untuk setiap stage dan panah merupakan harga yang dilalui dari state periode sebelum ke state periode setelahnya. Dan panah yang berwarna merah merupakan solusi untuk harga termurah mulai dari inisial status hingga *final* status untuk penyelesaian *forward dynamic programming*.

### 1.4.3 Lagrangian Relaxation

Pada metode *lagrangian relaxation* nilai inisial lambda akan ditentukan terlebih dahulu. Tentunya nilai dari lambda pertama bukanlah hasil yang benar. Ketika nilai total dari  $P_1 + P_2 + P_3 + \dots P_i < P_{target}$  maka nilai  $\lambda$  untuk iterasi berikutnya akan bertambah lebih besar dari nilai  $\lambda$  sebelumnya. Dan sebaliknya, jika nilai total  $P_1 + P_2 + P_3 + \dots P_i > P_{target}$  maka nilai  $\lambda$  untuk iterasi berikutnya akan lebih kecil daripada nilai dari  $\lambda$  sebelumnya. Proses ini akan melakukan itersi nilai  $\lambda$  hingga mendapatkan hasil dimana  $P_1 + P_2 + P_3 + \dots P_i = P_{target}$ .

Dasar dari metode *lagrangian relaxation* adalah pendekatan dual optimisasi dari *lagrange multiplier*. *Lagrange multiplier* menggunakan teknik yang menyelesaikan variabel lagrange, berupa lambda, dan setelahnya menemukan solusi dari variabel permasalahan yang ingin ditemukan.

*Objective function :*

$$\sum_{t=1}^T \sum_{i=1}^N [F_i(P_i^t) + Start\ up\ cost_{i,t}] U_i^t = F(P_i^t, U_i^t) \quad (2.5)$$

Keterangan :

$U_i^t$  : “on” atau “off” unit i saat t  
 $P_i^t$  : Daya yang dibangkitkan unit i saat t  
 $Start\ up\ cost_{i,t}$  : Biaya untuk menyalakan unit i saat t

Pada solusi *lagrangian relaxation* ada tiga hal lain yang harus diperhatikan setelah fungsi objektif yaitu :

1. Konstrain Beban

$$P_{load}^t - \sum_{i=1}^N P_i^t U_i^t = 0 \text{ for } t = 1 \dots T \quad (2.6)$$

2. Batas Unit

$$U_i^t P_i^{min} \leq P_i^t \leq U_i^t P_i^{max} \text{ for } i = 1 \dots N \text{ and } t = 1 \dots T \quad (2.7)$$

3. Unit minimum up- and down-time

Fungsi objektif pada persamaan (2.5) dapat di ubah menjadi persamaan lagrange dengan memperhatikan *constraint* yang ada untuk menemukan solusi dari *economic dispatch*.

$$\mathcal{L}(P, U, \lambda) = F_i^t(P_i^t) + \sum_{t=1}^T \lambda^t (P_{load}^t - \sum_{i=1}^N P_i^t U_i^t) \quad (2.8)$$

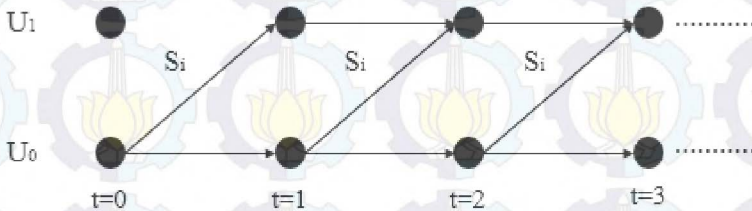
Dengan metode *lagrangian relaxation* kita mendapatkan solusi penyelesaian dari setiap unit secara terpisah, tidak memperhatikan unit lain. Persoalan *unit commitment* mengharuskan kita untuk meminimisasi fungsi lagrange persamaan (2.8). Prosedur *lagrangian relaxation* dilakukan melalui dengan prosedur *dual optimization*. Prosedur *dual optimization* dipaksa untuk mendapatkan hasil optimum dengan cara memaksimalkan fungsi lagrange dengan memperhatikan *lagrangian multipliers*, saat ingin meminimisasi variabel lain.

$$q(\lambda) = \min \mathcal{L}(P, U, \lambda) \quad (2.9)$$

$$q^*(\lambda) = \max q(\lambda) \quad (2.10)$$

Dengan persamaan (2.8), fungsi objektif setiap unit diselesaikan secara terpisah, tanpa memperhatikan fungsi objektif lain.

Ada dua *constraint* yang dimasukkan ke dalam persamaan (2.5) yaitu persamaan (2.7) dan minimum *up-* dan *down-time*. Untuk *constraint* minimum *up-* dan *down-time* dapat diselesaikan dengan *dynamic programming*, kemungkinan unit  $U_i^t$  hanya bernilai 1 dan 0, saat  $U_i^t=0$  maka unit tidak dioptimisasi dan saat  $U_i^t=1$  maka fungsi objektif unit i di optimisasi, seperti ditunjukkan pada Gambar 2.7.



**Gambar 2.9** Jalur Unit i untuk minimum *up- and down-time*



## 2.5 Economic Dispatch

*Economic dispatch* merupakan salah satu aspek yang perlu diperhatikan untuk mendapatkan biaya pembangkitan yang optimal. *Economic dispatch* ialah perhitungan output daya setiap pembangkit untuk beroperasi secara ekonomis pada kondisi beban setiap periode. ED merupakan optimasi *non-linear* yang variabel outputnya adalah keluaran daya pada setiap unit yang aktif. Setiap unit pembangkit mempunyai karakteristik fungsi biaya terhadap daya yang berbeda-beda, yang dipengaruhi oleh efisiensi unit pembangkit dan jenis bahan bakar yang digunakan. Secara matematis fungsi biaya adalah sebagai berikut :

$$\sum_{i=1}^n H_i = a_i P_i^2 + b_i P_i + c_i \quad (2.11)$$

$$\sum_{i=1}^n F_i = \sum_{i=1}^n f_i \times \sum_{i=1}^n H_i \quad (2.12)$$

$$\begin{aligned} F_T &= F_1 + F_2 + F_3 + \dots + F_n \\ &= \sum_{i=1}^n F_i \end{aligned} \quad (2.13)$$

Keterangan :

$H_i$	: Input kalor per unit (Btu/h or Mbtu/h)
$P_i$	: Daya operasi setiap unit pembangkit
$a_i$	: Koefesien orde-2 setiap unit
$b_i$	: Koefesien orde-1 setiap unit
$c_i$	: Koefesien <i>No-Load Cost</i> setiap unit
$F_i$	: Biaya yang diperlukan untuk membangkitkan $P_i$
$f_i$	: <i>Fuel Cost</i> (R per Btu / R per Mbtu)
$F_T$	: Total keseluruhan biaya semua unit

Secara matematis persamaan *economic dispatch* (2.12) dapat dipersingkat ke persamaan (2.13). Sistem yang di *dispatch* terdiri dari N unit pembangkit termal yang terhubung ke *single bus-bar* yang menyediakan daya untuk  $P_{load}$ .

## 2.6 Emission Dispatch

Saat beroperasi pembangkit listrik thermal, berbahan bakar fosil, akan menghasilkan emisi sebagai *side effect*. Emisi yang dihasilkan setiap unit pembangkit dapat dimodelkan secara matematis, sebagai berikut :

$$\sum_{i=1}^n E_i = \alpha_i P_i^2 + \beta_i P_i + \gamma_i \quad (2.14)$$

Keterangan :

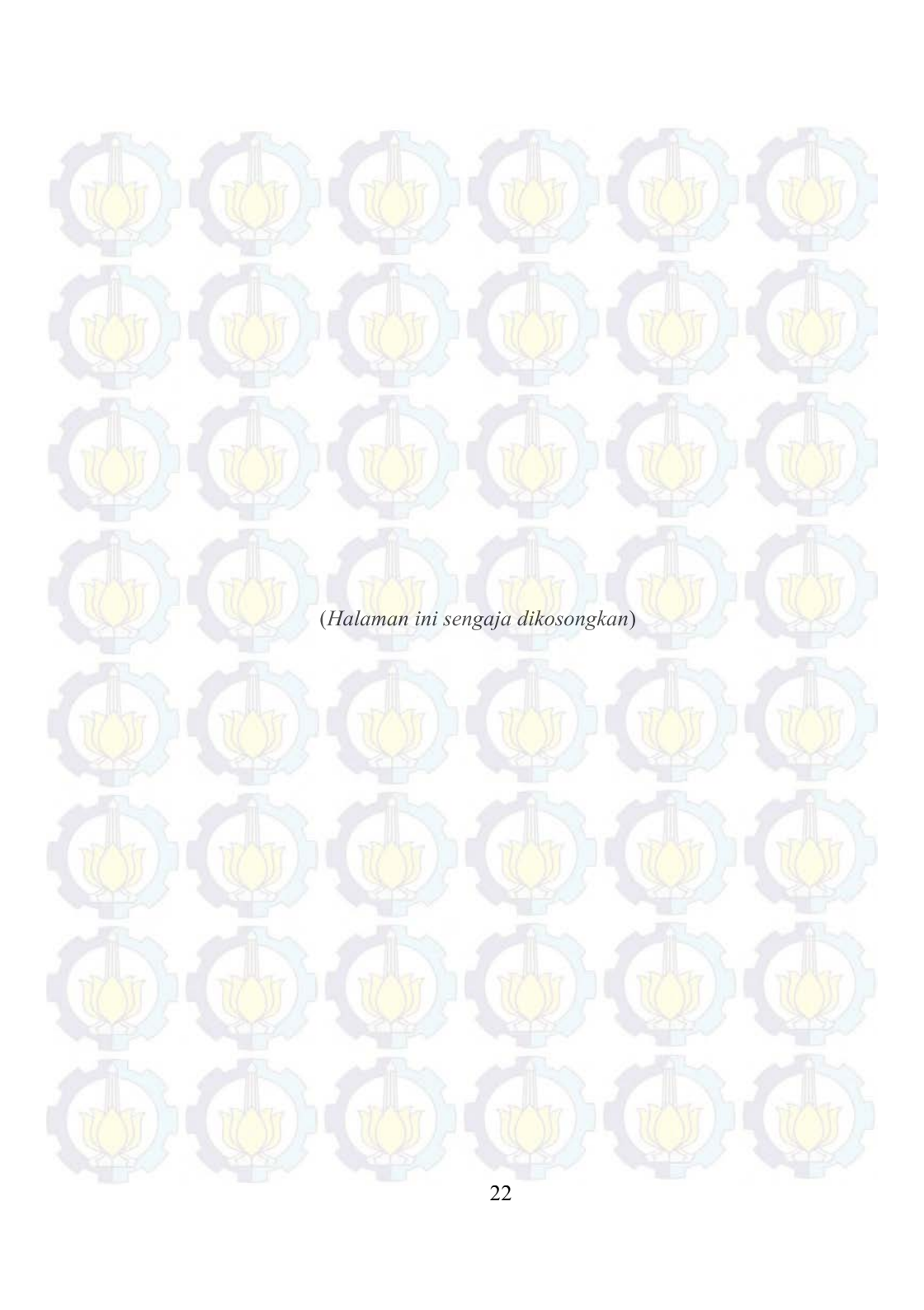
$E_i$  : Emisi (kg/h)

$P_i$  : Daya operasi setiap unit pembangkit

$\alpha_i$  : Koefesien emisi

$\beta_i$  : Koefesien emisi

$\gamma_i$  : Koefesien emisi



*(Halaman ini sengaja dikosongkan)*



## BAB 3

### IMPLEMENTASI *MULTIOBJECTIVE UNIT COMMITMENT* DENGAN METODE *LAGRANGIAN RELAXATION*

#### 3.1 *Multiobjective Optimization*

*Multiobjective Optimization* merupakan penyelesaian dari beberapa fungsi tujuan. Untuk menyelesaikan fungsi-fungsi tujuan yang ingin dicari terdapat banyak cara, salah satunya dengan mengkonversikan fungsi-fungsi tersebut menjadi satu, lalu dengan fungsi baru tersebut dicari nilai optimal pada fungsi baru. Dan setelah didapatkan *variabel* yang optimal pada fungsi baru, *variabel* yang didapatkan, dimasukkan ke dalam masing-masing fungsi. Secara umum bentuk matematis untuk *multiobjective optimization* adalah :

$$f(f_1(x), f_2(x), \dots, f_i(x)) \quad (3.1)$$

Dimana  $i$  merupakan jumlah fungsi yang ingin di gabung.

##### 3.1.1 *Price Penalty Factor* [6]

Pada tugas akhir ini fungsi baru yang ingin dibuat terdiri dari fungsi biaya dan fungsi emisi. *Economic dispatch* dan *emission dispatch* merupakan dua masalah yang berbeda. Jadi fungsi *economic* dan fungsi *emission* akan dikonversikan kedalam *single objective function* dengan menggunakan *price penalty factor*.

$$TC = FC(P_g) + h \times EC(P_g) \quad (3.2)$$

Dimana  $TC$  merupakan total biaya operasional pada sistem dengan *constraint* yang ada,  $FC$  merupakan fungsi biaya dan  $EC$  merupakan fungsi emisi. Dan  $h$  merupakan *price penalty factor*. *Price penalty factor* dapat di cari dengan langkah sebagai berikut :

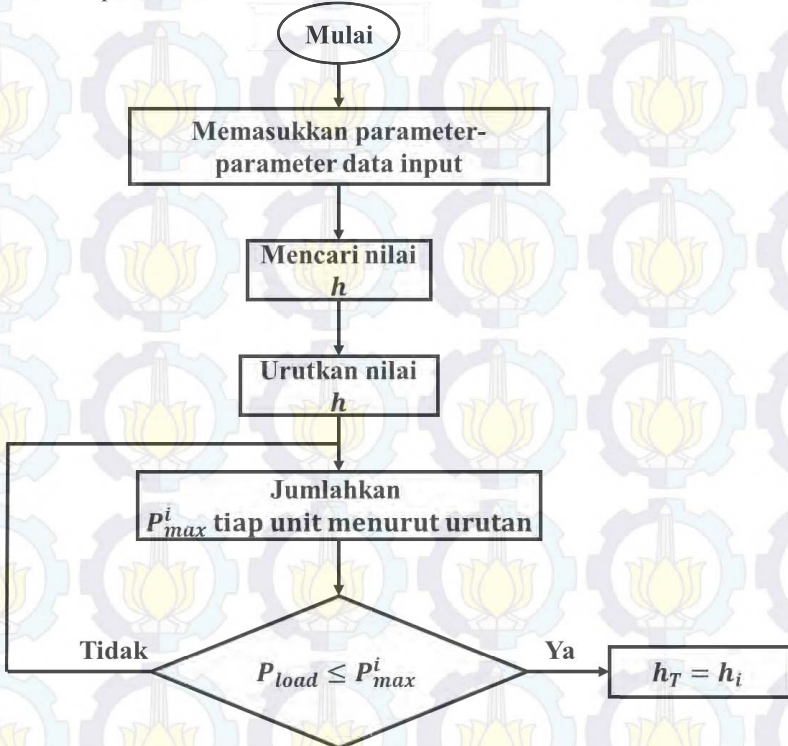
1. Hitung biaya pembangkitan unit pembangkit pada daya maksimal.
2. Hitung jumlah emisi unit pembangkit pada daya maksimal.
3. Bagi biaya pembangkitan dengan jumlah emisi pada setiap unit pembangkit.

$$h_i = \frac{FC_i(P_{\max})}{EC_i(P_{\max})}, h_i = 1, 2, 3, \dots, n \text{ R/kg} \quad (3.3)$$

4. Urut nilai *price penalty factor* mulai dari yang terkecil hingga terbesar.
5. Jumlahkan semua pembangkitan maksimal pada setiap unit mengikuti urutannya hingga

$$P_{\text{load}}^t \leq P_{\max}^i \quad (3.4)$$

6. Setelah memenuhi pada persamaan (3.4) maka nilai  $h_i$  yang akan digunakan untuk  $P_{\text{load}}^t$  adalah perhitungan  $h_i$  dengan penambahan unit terakhir.



Gambar 3.1 Diagram alur price penalty factor

### 3.1.2 Bobot

Pada masalah *multiobjective optimization*, bobot merupakan indikasi akan dibawa kemana penyelesaian multiobjektif tersebut. Pada *multiobjective unit commitment* bobot diletakkan kedalam persamaan emisi dan biaya dengan total semua bobot pada masing-masing persamaan bernilai 1. Secara matematis bobot dapat di rumuskan sebagai berikut :

$$\varphi_c(P_i) = w_{fuel}F_c(P_i) + w_{emi}E_c(P_i) \quad (3.5)$$

Untuk perhitungan data pada bobot terdapat tiga kasus yaitu

1. *Economic Dispatch*

Pada *Economic Dispatch*  $w_{fuel} = 1$  dan  $w_{emi} = 0$

$$\begin{aligned} \varphi_c(P_i) &= (1)F_c(P_i) + (0)E_c(P_i) \\ \varphi_c(P_i) &= F_c(P_i) \end{aligned} \quad (3.6)$$

Dari persamaan (3.6) maka fungsi yang akan diperhatikan adalah fungsi ekonomi saja, sehingga hasil yang akan didapatkan lebih cenderung untuk optimal pada sisi biaya saja, sedangkan pada sisi emisi lebih cenderung belum optimal.

2. *Emission Dispatch*

Pada *Emission Dispatch*  $w_{fuel} = 0$  dan  $w_{emi} = 1$

$$\begin{aligned} \varphi_c(P_i) &= (0)F_c(P_i) + (1)E_c(P_i) \\ \varphi_c(P_i) &= E_c(P_i) \end{aligned} \quad (3.7)$$

Dari persamaan (3.7) maka fungsi yang akan diperhatikan adalah fungsi emisi saja, sehingga hasil yang akan didapatkan lebih cenderung untuk optimal pada sisi emisi saja, sedangkan pada sisi ekonomi lebih cenderung belum optimal.

3. *Combined Economic Emission Dispatch*

Untuk kombinasi *economic* dan *emission dispatch* maka  $w_{fuel} = 1$  dan  $w_{emi} = 1$



### 3.1.3 Algoritma Multiobjective

Alur perhitungan *multiobjective* pada aplikasi *software* dimulai dengan mengambil semua data input untuk dilakukan perhitungan untuk menjadikan dua *objective function* menjadi *single objective function*.

$$\begin{bmatrix} \text{coeff } a_i \\ \text{coeff } b_i \\ \text{coeff } c_i \end{bmatrix} = \begin{bmatrix} \text{coeff } a_i^{\text{economic}} \\ \text{coeff } b_i^{\text{economic}} \\ \text{coeff } c_i^{\text{economic}} \end{bmatrix} + h_i \begin{bmatrix} \text{coeff } a_i^{\text{emission}} \\ \text{coeff } b_i^{\text{emission}} \\ \text{coeff } c_i^{\text{emission}} \end{bmatrix} \quad (3.8)$$

Lalu dimasukkan ke dalam perhitungan bobot

$$\begin{bmatrix} \text{coeff } a_i \\ \text{coeff } b_i \\ \text{coeff } c_i \end{bmatrix} = w_{fuel} \begin{bmatrix} \text{coeff } a_i^{\text{economic}} \\ \text{coeff } b_i^{\text{economic}} \\ \text{coeff } c_i^{\text{economic}} \end{bmatrix} + w_{emission} \left( h_i \begin{bmatrix} \text{coeff } a_i^{\text{emission}} \\ \text{coeff } b_i^{\text{emission}} \\ \text{coeff } c_i^{\text{emission}} \end{bmatrix} \right) \quad (3.9)$$

Untuk nilai *economic tendency*

$$\begin{bmatrix} \text{coeff } a_i \\ \text{coeff } b_i \\ \text{coeff } c_i \end{bmatrix} = \begin{bmatrix} \text{coeff } a_i^{\text{economic}} \\ \text{coeff } b_i^{\text{economic}} \\ \text{coeff } c_i^{\text{economic}} \end{bmatrix} \quad (3.10)$$

Untuk nilai *emission tendency*

$$\begin{bmatrix} \text{coeff } a_i \\ \text{coeff } b_i \\ \text{coeff } c_i \end{bmatrix} = \begin{bmatrix} \text{coeff } a_i^{\text{emission}} \\ \text{coeff } b_i^{\text{emission}} \\ \text{coeff } c_i^{\text{emission}} \end{bmatrix} \quad (3.11)$$

## 3.2 Lagrangian Relaxation

Pada tugas akhir ini metode yang digunakan untuk menyelesaikan permasalahan ialah *lagrangian relaxation*. Dengan *lagrangian relaxation* hasil yang didapatkan dari permasalahan *unit commitment* tidak hanya pada penjadwalan unit setiap interval tapi juga pengoptimalan pembebanan pada masing-masing unit yang beroperasi setiap interval. *Lagrangian relaxation* merupakan penyelesaian dengan dua solusi yaitu *dynamic programming* dan *lagrange multiplier*.

Metode ini didasarkan pada pendekatan dual optimisasi. Prosedur dual optimisasi berusaha untuk mendapatkan konstrain optimum dengan

memaksimalkan persamaan lagrange dengan mematuhi *lagrange multiplier* saat meminimisasi variabel lain. Metode ini dasarnya ialah menemukan nilai  $\lambda^t$  optimum masing-masing interval. Setelah unit yang aktif telah ditentukan pada *lagrangian relaxation* dan *dynamic programming* maka  $\lambda^t$  sudah dapat dicari dengan ketentuan yang berlaku. Diberikan contoh perhitungan untuk melakukan *multiobjective unit commitment* dengan *lagrangian relaxation*. Contoh perhitungan menggunakan 3 unit dengan data *cost function*, *emission function*, *spinning reserve*, minimum *up-* dan *down-time*, dan *start-up cost*.

Pertama-tama didefinisikan,

$U_i^t = 0$  untuk unit  $i$  yang *off-line* saat interval  $t$

$U_i^t = 0$  untuk unit  $i$  yang *on-line* saat interval  $t$

Dengan konstrain dan fungsi obyektif pada bab 2 dapat dibentuk fungsi lagrange dengan cara yang sama pada masalah *combined economic emission dispatch*:

$$\mathcal{L}(P, U, \lambda) = F(P_i^t, U_i^t) + \sum_{i=1}^T \lambda^t \left( P_{load}^t - \sum_{i=1}^N P_i^t U_i^t \right) \quad (3.12)$$

$$q(\lambda) = \min \mathcal{L}(P, U, \lambda) \quad (3.13)$$

$$q^*(\lambda) = \max q(\lambda) \quad (3.14)$$

Prosedur *lagrangian relaxation* menyelesaikan *unit commitment* dengan mengabaikan konstrain pada persamaan *coupling constraint*. Secara sederhana *lagrangian relaxation* dilakukan dengan dua langkah sederhana:

1. Temukan nilai  $\lambda^t$  setiap interval yang membuat nilai *duality gap* memenuhi persyaratan.
2. Diasumsikan nilai  $\lambda^t$  telah didapatkan pada langkah pertama, temukan nilai minimum  $\mathcal{L}$  dengan nilai  $P^t$  dan  $U^t$  yang telah diatur.

$$\min q(\lambda) = \sum_{i=1}^N \min \sum_{t=1}^T \{ [F_i(P_i^t + \text{Start up cost}_{i,t})] U_i^t - \lambda^t P_i^t U_i^t \} \quad (3.15)$$

Subject to

$$U_i^t P_i^{min} \leq P_i^t \leq U_i^t P_i^{max}$$

Dan pada konstrain *up*- dan *down-time* diselesaikan dengan *dynamic programming*. Bila  $U_i^t = 0$  maka unit tersebut diabaikan, dan bila  $U_i^t = 1$  maka fungsi unit tersebut harus diminimisasi ditunjukkan pada persamaan(3.16).

$$\min[F_i(P_i) - \lambda^t P_i^t] \quad (3.16)$$

Nilai minimum dari persamaan (3.16) didapatkan dari turunan pertama persamaan (3.16).

$$\frac{d}{dP_i^t} [F_i(P_i) - \lambda^t P_i^t] = \frac{d}{dP_i^t} F_i(P_i^t) - \lambda^t = 0 \quad (3.17)$$

Dan solusi dari persamaan (3.17) adalah

$$\frac{d}{dP_i^t} F_i(P_i^{opt}) = \lambda^t \quad (3.18)$$

Untuk penyelesaian pada persamaan (3.18) terdapat tiga hal yang menentukan hasil dari  $P_i^{opt}$ , sebagai berikut :

1. Jika  $P_i^{opt} \leq P_i^{min}$ , maka

$$\min[F_i(P_i) - \lambda^t P_i^t] = F_i(P_i^{min}) - \lambda^t P_i^{min} \quad (3.19)$$

2. Jika  $P_i^{min} \leq P_i^{opt} \leq P_i^{max}$ , maka

$$\min[F_i(P_i) - \lambda^t P_i^t] = F_i(P_i^{opt}) - \lambda^t P_i^{opt} \quad (3.20)$$

3. Jika  $P_i^{max} \leq P_i^{opt}$ , maka

$$\min[F_i(P_i) - \lambda^t P_i^t] = F_i(P_i^{max}) - \lambda^t P_i^{max} \quad (3.21)$$

Perlu diketahui bahwa kita meminimisasi persamaan  $[F_i(P_i) - \lambda^t P_i^t]$  pada setiap *stage* dan pada saat  $U_i^t = 0$  maka persamaan tersebut



bernilai nol, maka cara untuk meminimisasikan persamaan tersebut adalah

$$[F_i(P_i) - \lambda^t P_i^t] < 0 \quad (3.22)$$

Dengan semua persamaan *lagrangian relaxation* yang dijelaskan diatas semua permasalahan tersebut diselesaikan dengan nilai lambda yang telah didapatkan untuk penyelesaian permasalahan yang optimal. Pada *lagrangian relaxation* nilai lambda dicari dengan menggunakan *dual optimization*.

$$\lambda^t = \lambda^t + \left[ \frac{d}{d\lambda} q(\lambda) \right] \alpha \quad (3.23)$$

Setiap pencarian  $\lambda^t$  dilakukan secara terpisah. Nilai  $\alpha$  pada pencarian lambda harus ditentukan dengan benar agar nilai lambda mencapai nilai optimalnya. Jadi, apabila total daya yang dibangkitkan  $P_i^t$  tidak memenuhi *demand load* atau  $P_{load}^t$  dengan selisih nilai kurang dari 0 maka nilai lambda akan menggeser ke kanan menuju ke *stationery point*. Lalu apabila total daya yang dibangkitkan  $P_i^t$  tidak memenuhi *demand load* atau  $P_{load}^t$  dengan selisih nilai lebih dari 0 maka nilai lambda akan menggeser ke kiri mendekati *stationery point*.

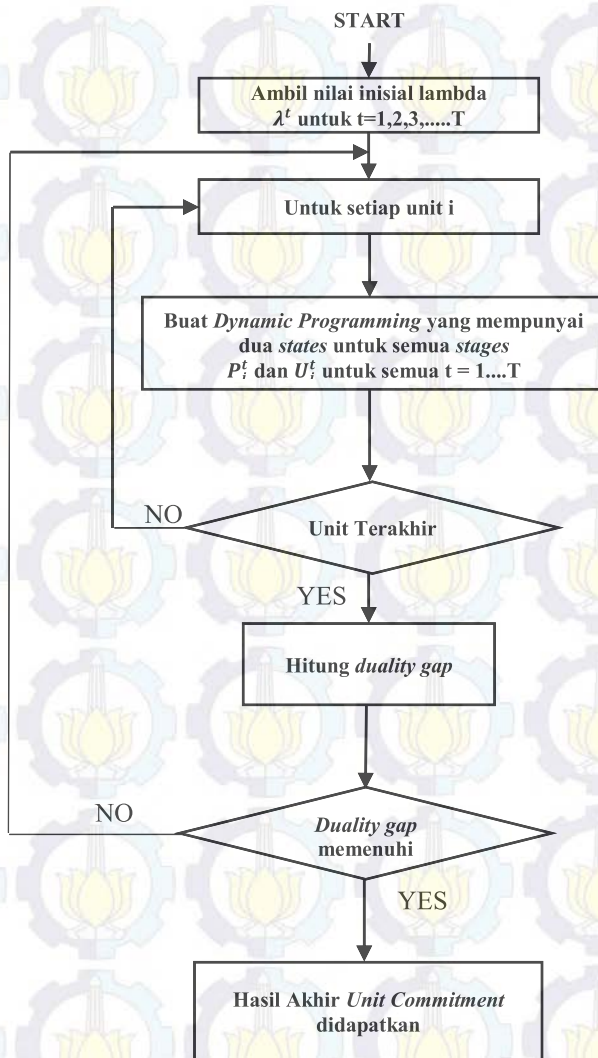
Di unit commitment metode *lagrangian relaxation* menggunakan iterasi dari algoritma *dynamic programming* untuk mendapatkan nilai awal yang baik untuk melakukan iterasi dual optimisasi. Dan pada akhir solusi *lagrangian relaxation*, logika *dynamic programming* akan mendapatkan solusi akhir dari permasalahan *unit commitment* menggunakan *lagrangian relaxation*.

Dimana lambda vektor akan bergeser ke kanan dan ke kiri, dengan syarat persamaan (3.24) dan persamaan (3.25)

$$\alpha = 0.01 \text{ ketika } \frac{d}{d\lambda} q(\lambda) \text{ bernilai positif} \quad (3.24)$$

$$\alpha = 0.002 \text{ ketika } \frac{d}{d\lambda} q(\lambda) \text{ bernilai negatif} \quad (3.25)$$

Dengan setiap interval nilai lambda akan diperlakukan secara terpisah.



**Gambar 3.2** Diagram alur *lagrangian relaxation*

### 3.3 Argumen *Input Multiobjective Unit Commitment*

Dari semua data yang diolah untuk mendapatkan hasil *unit commitment* pada *software* Delphi, data *input* tersebut akan dibuat kedalam argumen. Nantinya argumen *input* tersebut akan dimasukkan kedalam perhitungan. Argumen *input* yang digunakan dapat dilihat sebagai berikut:

**Tabel 3.1** Argumen *Input*

Argumen	Keterangan
CoeffEconomic [i]	Sebagai inputan awal dari nilai coefisien biaya a, b, c dalam persamaan $H_i = a_i P_i^2 + b_i P_i + c_i$
CoeffEmission [i]	Sebagai inputan awal dari nilai coefisien emisi $\alpha$ , $\beta$ , $\gamma$ dalam persamaan $E_i = \alpha_i P_i^2 + \beta_i P_i + \gamma_i$
Unitmax[i]	Sebagai inputan awal dari batas maximum pembangkitan unit (Pmax)
Unitmin[i]	Sebagai inputan awal dari batasan minimum pembangkitan unit (Pmin)
Fuelcost[i]	Sebagai inputan awal nilai dari fuelcost yang digunakan untuk persamaan $Fi(P_i(t)) = Hi(P_i(t)) \times fuelcost\ i$
Minimum up-time[i]	Sebagai inputan untuk batasan setiap unit i mempunyai waktu nyala minimal.
Minimum down-time[i]	Sebagai inputan untuk batasan setiap unit i mempunyai rentang waktu tidak boleh dinyalakan kembali.
SPR[i]	Sebagai inputan <i>spinning reserve</i> pada setiap unit i untuk semua interval $P_{i\ min}^t \leq P_{i\ generate}^t + SPR_i \leq P_{i\ max}^t$
Number of Interval	Jumlah interval yang akan dimasukkan kedalam perhitungan <i>unit commitment</i> .
Load Interval	Sebagai inputan nilai beban ( <i>Pload</i> ) di setiap interval.



### 3.4 Sintaksis *Software Multiobjective Unit Commitment* dengan Delphi

Sintaksis *software* adalah perintah yang digunakan untuk melakukan pemanggilan program dengan argumen input yang kita masukkan. Sintaksis yang digunakan pada *software* Delphi adalah sebagai berikut :

**Tabel 3.2** Sintaksis

Sintaksis	Keterangan
datadump	Digunakan sebagai memasukkan semua data awal perhitungan di setiap periode
Sort	Merupakan prosedur untuk mengurutkan nilai <i>price penalty factor</i> dari yang terkecil hingga terbesar.
ihr_ftn	Sebagai inisiasi turunan pertama persamaan $\varphi_c(P_i)$
EDC	Sebagai prosedur untuk mencari nilai $P_i^{opt}$ untuk setiap nilai lambda yang didapatkan.
lambda_search	Sebagai prosedur penjalanan menentukan nilai lambda yang digunakan dalam persamaan, yaitu dengan menentukan nilai $\lambda_{start}$ dan $\frac{d}{d\lambda}q(\lambda)$ untuk proses iterasi. Hasil akhir bertujuan untuk mendapatkan jumlah nilai <i>duality gap</i> yang bernilai $\leq 0.005$
FEAS	Prosedur untuk menentukan unit i aktif atau tidak untuk semua <i>stage</i>
BESTSTATE	Sebagai prosedur penentuan apakah unit tersebut merupakan <i>state</i> terbaik dengan memperhatikan <i>start-up cost</i> dan minimum <i>up- and down-time</i>
prod_cost	Mendapatkan nilai biaya pembangkitan setiap unit setelah mendapatkan nilai pembangkitan yang optimal dari proses <i>lagrangian relaxation</i>
Final Output	Prosedur <i>print</i> untuk menampilkan hasil akhir dari <i>multiobjective unit commitment</i> .

### 3.5 Penggunaan *software* untuk perhitungan *Multiobjective Unit Commitment*.

The screenshot displays the 'NewUnitComFile.Dat' window. At the top, it says 'UNIT COMMITMENT' and 'Data Generator'. Below this is a table with the following headers: Unit Number, Min MW, Max MW, EC1, EC2, EC3, EM2, EM3, and Start Up Cost. The table is currently empty. Below the table, there is a section for 'Number Of Units' with a value of 0. Below that, there is a section for 'Number Of Intervals' and 'Length Interval (Hour)'. There are also radio buttons for 'Spinning Reserve per Unit' (Yes/No) and 'Calculate Weight' (EconomicDispatch, EmissionDispatch, CombinedEmissionD, CombinedEconomicD). A 'Process' button is located at the bottom right.

**Gambar 3.3** *Main Display Multiobjective Unit Commitment*

Gambar 3.3 menunjukkan tampilan utama dari *software* perhitungan *multiobjective unitcommitment*. Pada *windows* utama MUC, data *input* dimasukkan kedalam masing-masing *subject*. Pada kolom data generator terdapat data *input* setiap unit dengan berdasarkan jumlah unit yang diminta. Untuk memasukkan data input dibutuhkan media untuk tempat memasukkan input. Jika pada data masukkan di klik *double* maka akan muncul *window* baru yang akan menuju *editor input* tiap unit untuk memasukkan data *input*. Gambar 3.4 merupakan tampilan yang digunakan untuk memasukkan data input.

Terdapat juga kolom untuk memasukkan data beban maksimal untuk 24 interval, kolom ditunjukkan pada gambar 3.5.

**Gambar 3.4** Editor Input

	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5

**Gambar 3.5** Input untuk beban setiap interval

Gambar 3.6 merupakan kolom *input* untuk memasukkan data beban setiap *interval*. Jumlah data beban yang diberikan sesuai dengan jumlah *interval* yang dimasukkan, dengan batasan maksimal sebanyak 24 *interval*.

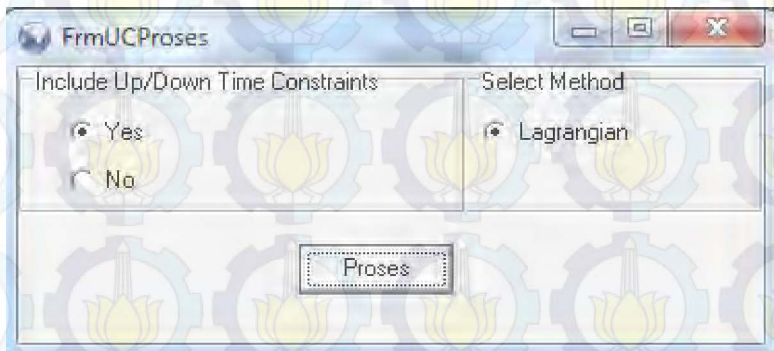
**Gambar 3.6** Weight Preferences Option perhitungan pada *software*



Pada kolom *calculate weight* akan ditentukan hasil *unit commitment* akan lebih cenderung ke sisi biaya atau sisi emisi.

1. EconomicDispatch  
Bobot akan ditaruh seluruhnya pada fungsi biaya.
2. EmissionDispatch  
Bobot akan ditaruh seluruhnya pada fungsi emisi.
3. CombinedEED  
Menggunakan *price penalty factor* untuk *composite* kedua fungsi menjadi *single objective function*. Dan pada *option* ini *price penalty factor* diletakkan pada fungsi emisi

Gambar 3.7 menunjukan metode yang akan ingin dipilih untuk penyelesaian *unit commitment* dan *option* untuk memilih memperhitungkan minimum *up- and down-time* atau tidak.



**Gambar 3.7** Display untuk *min.up- dan down-time option* dan *method options*



*(Halaman ini sengaja dikosongkan)*

## BAB 4 HASIL DAN SIMULASI

Pada bab ini akan diperlihatkan hasil perhitungan *multiobjective unit commitment* dengan menggunakan *lagrangian relaxation* dengan hasil unit “on” dan “off” dan juga disertakan hasil pembebanan setiap unit. Hasil perhitungan yang diperlihatkan berupa tabel-tabel yang meliputi pembangkitan yang optimal, biaya total, emisi total dan penjadwalan pembangkit. Hasil dari simulasi ini akan memperlihatkan perbandingan dari aplikasi metode untuk beberapa kasus. Data yang dimasukkan ke dalam perhitungan adalah data IEEE 39 bus. Berikut merupakan data yang dimasukkan dalam simulasi :

**Tabel 4.1** Data masukan *cost function* IEEE 39 bus

Unit	Pmin (MW)	Pmax (MW)	$a_i$ (\$/MWh <sup>2</sup> )	$b_i$ (\$/MWh)	$c_i$ (\$)
1	150	455	0,0004	16,19	1000
2	150	455	0,0003	17,26	970
3	20	130	0,002	16,6	700
4	20	130	0,0021	16,5	680
5	25	162	0,0039	19,7	450
6	20	80	0,0071	22,36	370
7	25	85	0,0079	27,74	480
8	10	55	0,0022	27,27	665
9	10	55	0,0017	27,79	670
10	10	55	0,0041	25,92	660

**Tabel 4.2** Data masukan *emission function* IEEE 39 bus

Unit	$\alpha$ (ton/MWh <sup>2</sup> )	$\beta$ (ton/MWh)	$\gamma$ (ton)
1	0,00312	-0,24444	10,33908
2	0,00312	-0,24444	10,33908
3	0,00509	-0,40695	30,0391
4	0,00509	-0,40695	30,0391



**Tabel 4.2** Data masukan *emission function* IEEE 39 bus (lanjutan)

Unit	$\alpha(\text{ton/MWh}^2)$	$\beta(\text{ton/MWh})$	$\gamma(\text{ton})$
5	0,00344	-0,38132	32,00006
6	0,00344	-0,38132	32,00006
7	0,00465	-0,39023	33,00056
8	0,00465	-0,39254	35,00056
9	0,0047	-0,39864	36,00012
10	0,00465	-0,39023	33,00066

**Tabel 4.3** Data masukan minimum *up- and down-time, start-up cost, dan* inisial status

Unit	Up-Time (h)	Down Time(h)	Start-up Cost (\$)	Initial Status (h)
1	8	8	4500	8
2	8	8	5000	8
3	5	5	550	-5
4	5	5	560	-5
5	6	6	900	-6
6	3	3	170	-3
7	3	3	260	-3
8	1	1	30	-1
9	1	1	30	-1
10	1	1	30	-1

## 4.1 Studi Kasus

### 4.1.1 Kasus 1

Pada studi kasus 1 diperlihatkan hasil perhitungan yang hanya memperhatikan dari sisi biaya bahan bakar BEC(*Best Cost Commitment*). Pada BEC  $w_{fuel} = 1$  dan  $w_{emission} = 0$ , maka  $\varphi_c(P_i) = F_c(P_i)$  dimana  $F_c$  merupakan *cost function*.

**Tabel 4.4** Hasil *running* aplikasi *PowerGen* (BCC) Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
4	OFF	OFF	OFF	ON	ON	ON	ON	ON
5	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
6	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
8	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
9	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
10	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
Cost(\$)	13687	14527	16302	18590	19473	21738	22856	23901
Emission(ton)	684,3	753,9	947,9	941,6	1052	1175	1185	1212
Load(MW)	700	750	850	950	1000	1100	1150	1200

**Tabel 4.5** Hasil *running* aplikasi *PowerGen* (BCC) Interval 9-16

Unit \ Interval	9	10	11	12	13	14	15	16
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	OFF	OFF
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	ON	ON
6	OFF	ON	ON	ON	ON	OFF	OFF	OFF
7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
8	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF
9	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
10	OFF	OFF	ON	ON	OFF	OFF	OFF	OFF
Cost(\$)	26161	28752	30686	32600	28752	26161	23901	20847

**Tabel 4.5** Hasil *running* aplikasi *PowerGen*(BCC) Interval 9-16 (lanjutan)

Unit \ Interval	9	10	11	12	13	14	15	16
Emission(ton)	1257	1299	1325	1351	1299	1257	1212	1135
Load(MW)	1300	1400	1450	1500	1400	1300	1200	1050

**Tabel 4.6** Hasil *running* aplikasi *PowerGen* (BCC) Interval 17-24

Unit \ Interval	17	18	19	20	21	22	23	24
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	OFF	OFF	OFF	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	OFF	OFF	OFF
5	ON	ON	ON	ON	ON	OFF	OFF	OFF
6	OFF	OFF	OFF	ON	ON	ON	OFF	OFF
7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
8	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
9	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
10	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
Cost(\$)	19998	21838	23901	28752	26574	21964	17782	16027
Emission(ton)	1022	1175	1212	1299	1274	1175	851,5	709,7
Load(MW)	1000	1100	1200	1400	1300	1100	900	800

Total biaya pada kasus 1 adalah sebesar 548930,96 \$ dan total emisi yang dihasilkan pada semua interval sebesar 26804,43 Ton.

#### 4.1.2 Kasus 2

Pada studi kasus 1 diperlihatkan hasil perhitungan yang hanya memperhatikan dari sisi ekonomi BEC(*Best Economic Commitment*). Pada BEC  $w_{fuel} = 0$  dan  $w_{emission} = 1$ , maka  $\varphi_c(P_i) = E_c(P_i)$  dimana  $E_c$  merupakan *emission function*.

**Tabel 4.7** Hasil *running* aplikasi *PowerGen* (BEC) Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
1	ON	ON	ON	ON	ON	ON	ON	ON



**Tabel 4.7** Hasil *running* aplikasi *PowerGen* (BEC) Interval 1-8 (lanjutan)

Unit \ Interval	1	2	3	4	5	6	7	8
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	ON	ON
6	ON	ON	ON	ON	ON	ON	ON	ON
7	ON	ON	ON	ON	ON	ON	ON	ON
8	OFF	OFF	ON	ON	ON	ON	ON	ON
9	OFF	OFF	OFF	ON	ON	ON	ON	ON
10	OFF	OFF	ON	ON	ON	ON	ON	ON
Cost(\$)	18088	19100	23189	26235	27161	28888	29717	30577
Emission(ton)	223,9	242	289	339	370	448	493	548
Load(MW)	700	750	850	950	1000	1100	1150	1200

**Tabel 4.8** Hasil *running* aplikasi *PowerGen* (BEC) Interval 9-16

Unit \ Interval	9	10	11	12	13	14	15	16
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	ON	ON
6	ON	ON	ON	ON	ON	ON	ON	ON
7	ON	ON	ON	ON	ON	ON	ON	ON
8	ON	ON	ON	ON	ON	ON	ON	ON
9	ON	ON	ON	ON	ON	ON	ON	ON
10	ON	ON	ON	ON	ON	ON	ON	ON
Cost(\$)	32255	33952	34803	35661	33952	32255	30577	28025
Emission(ton)	678	841	934	1.035	840,6	678	548	406

**Tabel 4.8** Hasil *running* aplikasi *PowerGen* (BEC) Interval 17-24 (lanjutan)

Unit \ Interval	9	10	11	12	13	14	15	16
Load(MW)	1300	1400	1450	1500	1400	1300	1200	1050

**Tabel 4.9** Hasil *running* aplikasi *PowerGen* (BEC) Interval 17-24

Unit \ Interval	17	18	19	20	21	22	23	24
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	OFF
4	ON	ON	ON	ON	ON	ON	ON	OFF
5	ON	ON	ON	ON	ON	ON	ON	OFF
6	ON	ON	ON	ON	ON	ON	ON	OFF
7	ON	ON	ON	ON	ON	ON	ON	OFF
8	ON	ON	ON	ON	ON	ON	ON	OFF
9	ON	ON	ON	ON	ON	ON	OFF	OFF
10	ON	ON	ON	ON	ON	ON	ON	OFF
Cost(\$)	27161	28888	30577	33952	32255	28888	24126	15466
Emission(ton)	370	448	548	841	678	448	314	824
Load(MW)	1000	1100	1200	1400	1300	1100	900	800

Total biaya pada kasus 2 adalah sebesar 688279,32 \$ dan total emisi yang dihasilkan pada semua interval sebesar 13383,21 Ton.

#### 4.1.3 Kasus 3

Pada studi kasus 3 diperlihatkan hasil perhitungan yang memperhatikan dua fungsi, fungsi biaya dan fungsi emisi, yang dibentuk kedalam *single objective function*.

**Tabel 4.10** Hasil *running* aplikasi *PowerGen* (*Combined Economic Emission*) Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON

**Tabel 4.10** Hasil *running* aplikasi *PowerGen(Combined Economic Emission)* Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
3	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
5	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
6	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON
7	OFF	OFF	OFF	OFF	ON	ON	ON	ON
8	OFF	OFF	ON	ON	ON	ON	ON	ON
9	OFF	ON	ON	ON	ON	ON	ON	ON
10	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON
Cost(\$)	13759	15877	18813	21675	22812	25360	26047	28224
Emission(ton)	614	633	748	869	922	991	1.031	1.125
Load(MW)	700	750	850	950	1000	1100	1150	1200

**Tabel 4.11** Hasil *running* aplikasi *PowerGen (Combined Economic Emission)* Interval 9-16

Unit \ Interval	9	10	11	12	13	14	15	16
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	OFF	OFF	ON	ON	OFF	ON	OFF
4	ON	OFF	ON	ON	OFF	OFF	OFF	OFF
5	OFF	ON	ON	ON	ON	ON	OFF	OFF
6	OFF	ON	ON	OFF	ON	ON	OFF	OFF
7	ON	ON	ON	ON	OFF	ON	ON	ON
8	ON	ON	ON	ON	ON	ON	ON	ON
9	ON	ON	ON	ON	ON	ON	ON	ON
10	OFF	ON	OFF	OFF	ON	OFF	ON	ON
Cost(\$)	29249	32616	32952	33640	31846	29751	28061	24828
Emission(ton)	1.141	1.282	1.255	1.295	1.198	1.143	1.045	936
Load(MW)	1300	1400	1450	1500	1400	1300	1200	1050



**Tabel 4.12** Hasil *running* aplikasi *PowerGen (Combined Economic Emission)* Interval 17-24

Unit \ Interval	17	18	19	20	21	22	23	24
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
4	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF
5	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF
6	ON	OFF	OFF	OFF	ON	OFF	OFF	OFF
7	ON	ON	ON	ON	ON	ON	OFF	OFF
8	ON	OFF	ON	ON	ON	ON	ON	OFF
9	OFF	ON	ON	ON	ON	ON	ON	ON
10	OFF	ON	ON	ON	ON	OFF	ON	ON
Cost(\$)	22388	25124	28029	32451	30574	25197	20826	17891
Emission (ton)	862	918,7	1045	1196	1115	918,5	763,2	648,8
Load(MW)	1000	1100	1200	1400	1300	1100	900	800

Total biaya pada kasus 3 adalah sebesar 627747,85 \$ dan total emisi yang dihasilkan pada semua interval sebesar 23695,47 Ton.

#### 4.1.4 Kasus 4

*Spinning Reserve* atau cadangan berputar merupakan hal yang perlu diperhatikan untuk menunjang keandalan kontinuitas sistem tenaga listrik. Pada kasus 4 ini diasumsikan pada masing-masing generator untuk memiliki *spinning reserve* agar sistem tetap beroperasi apabila terdapat salah satu unit yang *collapse*. Pada kasus 4 sistem akan diberikan *spinning reserve* sebesar daya maksimal salah satu unit yang terbesar, yaitu 455 MW, dan *spinning reserve* akan disebar secara merata pada seluruh unit dari kapasitas maksimal daya pembangkitan seluruh unit.

$$\begin{aligned}
 \text{SPR} &= \frac{455}{\sum_{i=1}^N P_i^{\max}} \times 100\% \\
 &= \frac{455}{1662} \times 100\% \text{ MW} \\
 &= 27,3 \text{ \% dari total seluruh pembangkitan}
 \end{aligned}$$

**Tabel 4.13** Hasil *running* aplikasi PowerGen (Combined Economic Emission with SPR) Interval 1-8

Unit \ Interval	1	2	3	4	5	6	7	8
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	OFF	OFF	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	OFF	OFF	OFF	ON	ON	ON	ON	ON
6	OFF	OFF	OFF	OFF	ON	ON	ON	ON
7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
8	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
9	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
10	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
Cost(\$)	14416	15265	17645	20170	21728	25532	27488	29515
Emission(ton)	481,3	567,3	613,8	616,2	623	707,2	752,5	757,5
Load(MW)	700	750	850	950	1000	1100	1150	1200

**Tabel 4.14** Hasil *running* aplikasi PowerGen (Combined Economic Emission with SPR) Interval 9-16

Unit \ Interval	9	10	11	12	13	14	15	16
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	ON	ON
6	ON	ON	ON	ON	ON	ON	ON	ON
7	ON	ON	ON	ON	ON	ON	ON	OFF
8	ON	ON	ON	ON	ON	ON	ON	OFF
9	ON	ON	ON	ON	ON	ON	ON	OFF
10	ON	ON	ON	ON	ON	ON	ON	ON
Cost(\$)	29515	29515	29515	29515	29515	29515	29515	23602
Emission(ton)	757,5	757,5	757,5	757,5	757,5	757,5	757,5	665

**Tabel 4.14** Hasil *running* aplikasi *PowerGen(Combined Economic Emission with SPR)* Interval 9-16 (lanjutan)

Unit \ Interval	9	10	11	12	13	14	15	16
Load(MW)	1200	1200	1200	1200	1200	1200	1200	1050

**Tabel 4.15** Hasil *running* aplikasi *PowerGen (Combined Economic Emission with SPR)* Interval 17-24

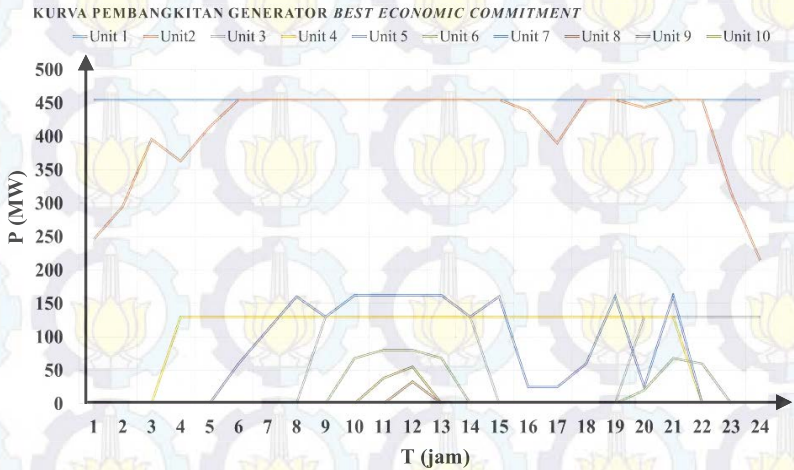
Unit \ Interval	17	18	19	20	21	22	23	24
1	ON	ON	ON	ON	ON	ON	ON	ON
2	ON	ON	ON	ON	ON	ON	ON	ON
3	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	ON
5	ON	ON	ON	ON	ON	ON	OFF	OFF
6	ON	ON	ON	ON	ON	ON	ON	OFF
7	OFF	OFF	ON	ON	ON	OFF	OFF	OFF
8	OFF	ON	ON	ON	ON	ON	OFF	OFF
9	OFF	OFF	ON	ON	ON	OFF	OFF	OFF
10	OFF	ON	ON	ON	ON	ON	OFF	OFF
Cost(\$)	21728	25532	29515	29515	29515	25532	19203	16798
Emission(ton)	623	707,2	757,5	757,5	757,5	707,2	620,1	526,9
Load(MW)	1000	1100	1200	1200	1200	1100	900	800

Dapat dilihat pada Tabel 4.14 untuk interval ke-9 hingga 15, interval 20 dan interval 21 melewati batasan maksimum dari kapasitas daya maksimum 10 unit pembangkit setelah diberi masukkan *spinning reserve*, maka untuk pembangkitan yang lebih besar dari 1200 MW seluruh unit tidak dapat beroperasi. Maka untuk semua rentang waktu pembebanan sistem beroperasi hanya pada interval 1 hingga 13 hingga 24. Salah satu solusinya adalah melakukan pemangkasan beban sehingga semua beban setiap interval yang melewati batasan maksimum akan dipangkas hingga 1200 MW. Dan total biaya seluruh operasi untuk 24 interval pada kasus 4 adalah sebesar 602184,28 \$ dan emisi yang dihasilkan sebesar 16542,66 ton.

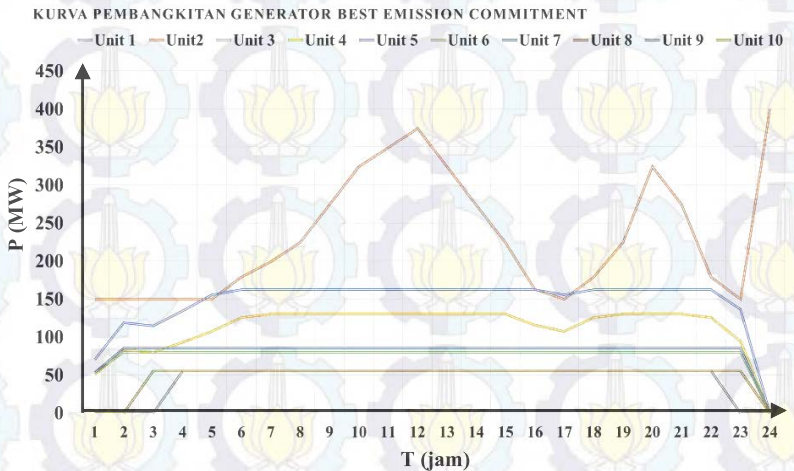
Seperti pada kasus 4 akan diberikan hasil dari interval beban seperti pada kasus 4 tetapi *spinning reserve* tidak dimasukkan kedalam perhitungan.



## 4.2 Kurva Daya untuk semua kasus.



Gambar 4.1 Kurva daya *Best Economic Commitment*



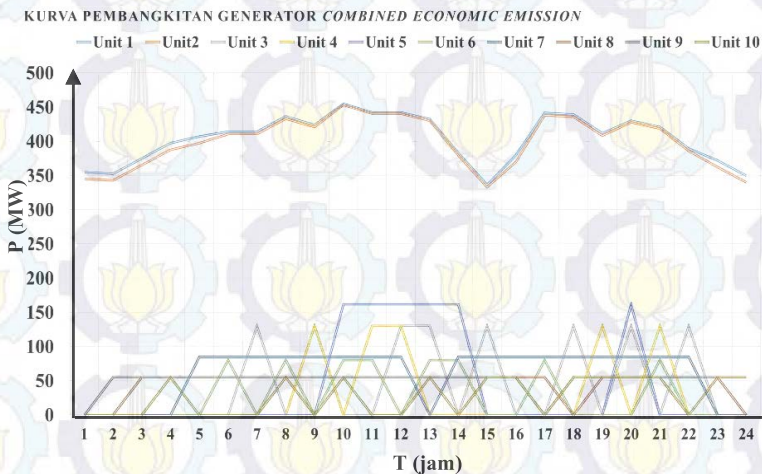
Gambar 4.2 Kurva daya *Best Emission Commitment*

Gambar 4.1 merupakan kurva daya terbangkit setiap generator pada 24 interval dengan rentang interval setiap 1 jam dengan pengoptimalan hanya pada fungsi biaya. Dan terlihat bahwa unit 1 dan unit 2 lebih

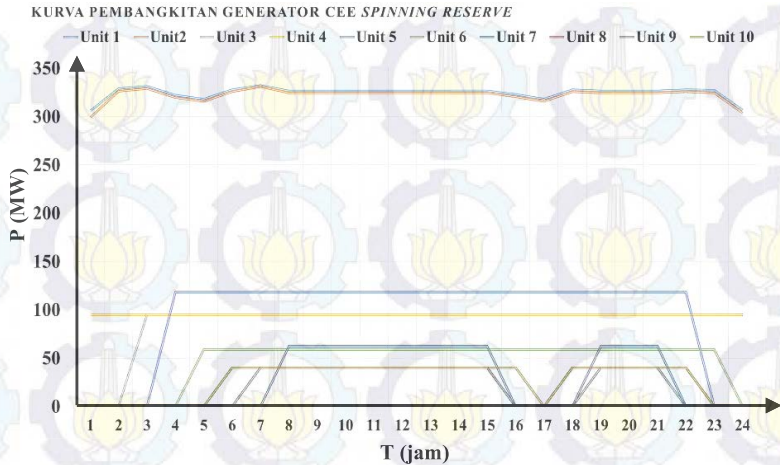
cenderung terbebani dengan beban yang lebih besar dikarenakan fungsi biaya unit dan unit 2 merupakan fungsi biaya yang lebih minimal daripada unit lainnya. Sehingga hasil yang didapatkan lebih cenderung ke sisi biaya tanpa memperhatikan emisi yang dihasilkan.

Pada Gambar 4.2 merupakan kurva daya terbangkit dengan hanya mengoptimalkan fungsi emisi. Pada gambar terlihat unit 1 dan unit 2 tidak dibebankan secara maksimal sedangkan pada unit lain ditekankan pada kapasitas daya maksimal. Sehingga didapatkan emisi yang paling optimal tetapi tidak memperhatikan dari sisi biaya.

Untuk memperhatikan biaya dan emisi yang dihasilkan maka diperlukannya *composite* kedua fungsi, fungsi biaya dan fungsi emisi. Kedua fungsi tersebut dibentuk kedalam *single objective function*, dan daya yang dibangkitkan setiap unit terlihat pada Gambar 4.3. hasil yang didapatkan biaya pada *combined economic emission* lebih mahal daripada *economic dispatch* tetapi lebih murah daripada *emission dispatch*. Sebaliknya emisi yang dihasilkan untuk pengoptimalan *combined economic emission* lebih besar daripada *emission dispatch* tetapi hasil emisinya lebih sedikit bila dibandingkan dengan *economic dispatch*.



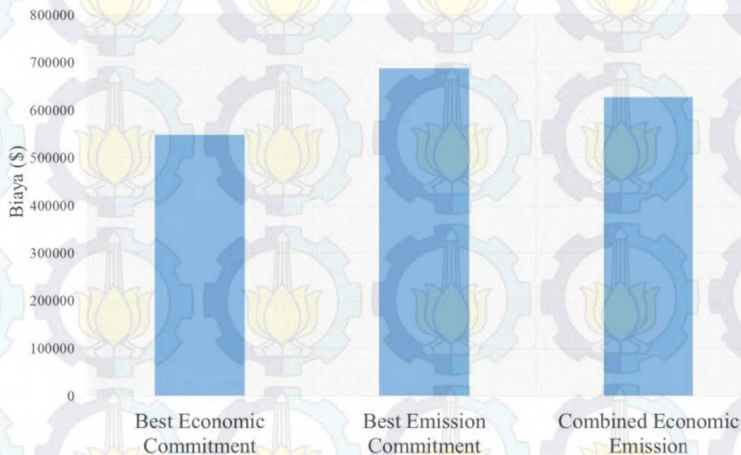
**Gambar 4.3** Kurva daya *Combined Economic Emission*



**Gambar 4.4** Kurva daya *Combined Economic Emission with Spinning Reserve*

### 4.3 Analisa Semua Kasus

Kasus pertama dan kedua dengan masing-masing memiliki kecenderungan ke sisi biaya bahan bakar dan sisi emisi, sehingga kedua kasus tersebut memiliki hasil yang cenderung untuk masing-masing sektor. Gambar 4.5 menunjukkan masing-masing hasil dari biaya yang dikeluarkan untuk memenuhi beban.



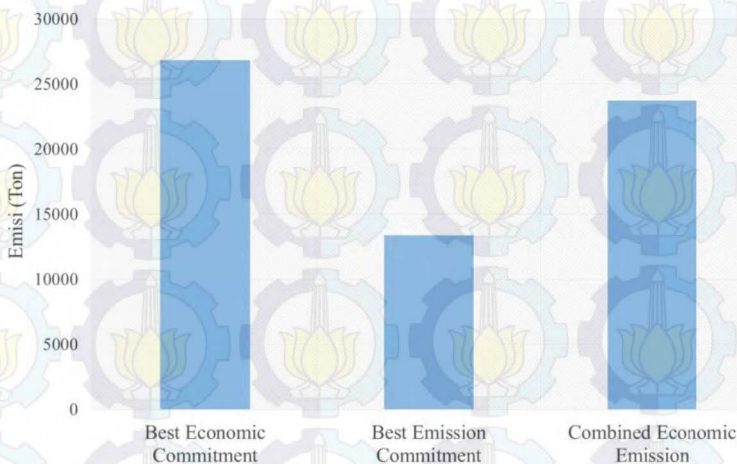
**Gambar 4.5** Grafik biaya untuk kasus 1, 2 dan 3



Biaya(BEC) = 548930,96 \$  
 Biaya(BEMC) = 688279,32 \$  
 Margin Biaya (Biaya-Emisi) = -139349,22 \$

Biaya pada kasus pertama lebih murah dikarenakan fungsi biaya untuk pembangkit dengan kapasitas yang terbesar, unit 1 dan unit 2, memiliki fungsi biaya yang paling minimal diantara semua pembangkit yang lain maka kedua unit tersebut selalu dibangkitkan secara maksimal bila mungkin untuk memenuhi *demand load*. Dan biaya *start-up cost* unit 1 dan unit 2 merupakan biaya yang terbesar diantara semua unit jika ditotal biaya *start-up cost* kedua unit tersebut sebesar 9500 \$, dibandingkan saja total biaya *start-up cost* delapan unit lain hanya sebesar 2500 \$ sehingga kedua unit tersebut sangat rugi bila di “off” dan “on” pada 24 interval tersebut mengingat minimum-up dan down time nya selama 8 jam. Sedangkan pada emisi unit 1 dan unit 2 menghasilkan emisi terbesar bila dibandingkan dengan unit lain untuk pembangkitan maksimal maka unit 1 dan unit 2 hanya dinyalakan untuk memenuhi beban.

Emisi(BEC) = 26804.43 Ton  
 Emisi(BEMC) = 13383,21 Ton  
 Margin Emisi (Biaya-Emisi) = 13421,22 Ton



**Gambar 4.6** Grafik emisi untuk kasus 1, 2 dan 3

Lain halnya pada emisi karena untuk *Best Emission Commitment* fungsi emisi pada unit 1 dan unit 2 bertolak belakang dengan fungsi biayanya dan oleh karena itu pada unit 1 dan unit 2 tidak pernah di setel ke kapasitas maksimumnya bila *demand load* nya sudah terpenuhi. Tetapi pada kasus 2, mengingat *minimum-up* dan *down time* yang tinggi untuk unit 1 dan unit 2 dan *demand load* yang harus dipenuhi maka unit 1 dan unit 2 selalu dinyalakan. Sehingga terlihat pada Gambar 4.6 nilai emisi yang dihasilkan pada kasus 1 sangatlah tinggi bila dibandingkan dengan kasus 2.

Pada kasus 3 mengombinasikan kedua fungsi, fungsi biaya dan fungsi emisi membuat kedua sektor biaya dan emisi tidak tindih sebelah.

$$\text{Biaya(Combined)} = 627747,85 \$$$

$$\text{Emisi(Combined)} = 23695,47 \text{ Ton}$$

$$\text{Margin Biaya(Combined-BEC)} = 78816,89 \$$$

$$\text{Margin Emisi(Combined-BEC)} = 3108,96 \text{ Ton}$$

$$\text{Margin Biaya(Combined-BEmC)} = 60531,47 \$$$

$$\text{Margin Emisi(Combined-BEmC)} = 10312,26 \text{ Ton}$$

Untuk kasus 4, menggunakan *spinning reserve* sebagai tambahan konstrain, mengakibatkan melonjaknya biaya pembangkitan dan emisi yang dihasilkan, dikarenakan untuk nilai yang dicapai belum mencapai optimal tetapi pembangkitan sudah pada batas maksimumnya akibat *spinning reserve*.

Dengan *Spinning Reserve*

$$\text{Cost} = 602184,28 \$$$

$$\text{Emission} = 16542,66 \text{ ton}$$

Tanpa *Spinning Reserve*

$$\text{Cost} = 573510,30 \$$$

$$\text{Emission} = 22180,34 \text{ ton}$$

$$\text{Margin Cost} = 602184,28 - 573510,30$$

$$= 28674,28 \$$$

$$\text{Margin Emission} = 16542,66 - 22180,34$$

$$= 5637,68 \text{ ton}$$



*(Halaman ini sengaja dikosongkan)*



## DAFTAR PUSTAKA

- [1] Wood, Allen J & Wollenberg, Bruce F, "*Power Generation, Operation, and Control*", John Wiley & Sons Inc, 1996.
- [2] NP, Padhy, "*Unit commitment – a bibliographical survey*". IEEE Trans Power System 2004;19(2):1196–205.
- [3] Raglend, I Jacob, "*Solutions to Practical Unit Commitment Problems with Operational, Power Flow and Enviromental Constraints*", 1-4244-0493-2/06 2006 IEEE
- [4] Sundaram, C Shanmuga, "*Solution to Profit Based Unit Commitment using Swarm Intelligence Technique*", 2014 ICCPEIC
- [5] Mariano, Silvio, "*Unit Commitment with Environmental Considerations: A Practical Approach*", 15<sup>th</sup> PSCC 2005
- [6] Mandal, K K, "*Non-Convex Emission Constrained Economic Dispatch Using Self-Adaptive Particle Swarm Optimization Technique*", 1568-4946 2014 Elsevier
- [7] W. L. Snyder, H. D. Powell, J. C. Payburn, "*Dynamic Programming Approach to Unit Commitment*," IEEE Transactions on Power Systems, vol. 2, No.2, pp. 339~350,1987
- [8] Benhamida, Farid, "*Enhanced lagrangian Relaxation Solution to The Generation Scheduling Problem*", Electrical Power and Energy Systems 32 (2010) 1099-1105
- [9] Ali Ahmadian, Mahdi Sedghi, Masoud Aliakbar Golkar, "*Solving Multiobjective Unit Commitment Problem to Minimize Operation Cost and Emissions Using HBMO Algorithm*", ICMLEME 2014

## BAB 5

### KESIMPULAN & SARAN

#### 5.1 Kesimpulan

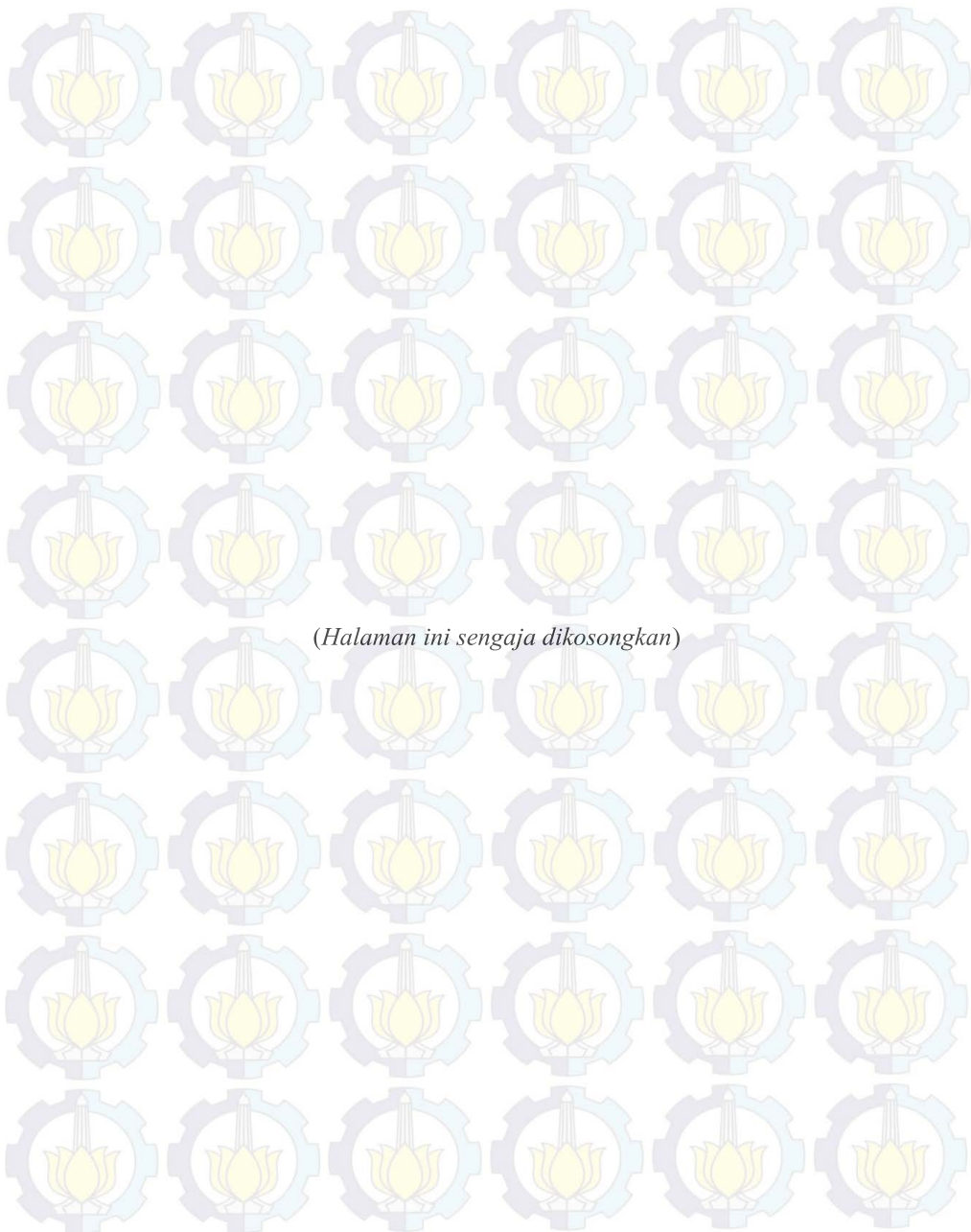
Dari hasil analisis *software* PowerGen pada *economic dispatch*, *emission dispatch* dan *combined economic emission* untuk perhitungan *multiobjective unit commitment* menggunakan *lagrangian relaxation* pada sistem kelistrikan:

1. Aplikasi perhitungan ini, PowerGen, mampu menentukan pembangkit yang harus dioperasikan dan tidak dioperasikan untuk mendapatkan pembebanan yang optimal setiap unit pembangkit.
2. Hasil yang diperoleh dari *software* PowerGen ini dapat memilih jenis pengoptimalan yang diinginkan, mulai dari sisi ekonomi, sisi emisi ataupun keduanya.
3. Hasil penjadwalan menggunakan *software* PowerGen belum menjamin untuk dapat diterapkan pada sistem interkoneksi karena adanya batasan-batasan yang belum dipenuhi.
4. Dengan adanya *spinning reserve* biaya operasi pada sistem akan semakin naik akibat pembatasan pembangkitan maksimum.

#### 5.2 Saran

Adapun saran untuk penelitian selanjutnya pada bidang operasi sistem tenaga berdasarkan hasil simulasi dan analisa pada tugas akhir ini, yaitu:

1. *Software* PowerGen dapat dikembangkan lebih lanjut untuk penambahan *transmission losses* pada perhitungan *multiobjective unit commitment*.
2. Adanya penambahan *ramp-rate* pada *software* perhitungan PowerGen.



*(Halaman ini sengaja dikosongkan)*



## LAMPIRAN

Berikut merupakan implementasi algoritma *visual basic* untuk *multiobjective unit commitment* :

```

interface
{
-----}
{
Unit Commitment Program from chapter 5
POWER GENERATION OPERATION AND CONTROL
}
{
Allen J. Wood and Bruce F. Wollenberg
}
-----}
CONST
INFINITY = 1.0E15;
MAX_NUMUNITS = 11;
MAX_NUMSTATES = 2048;
MAX_NUMPERIODS = 24;

{
-----}
{ <f> <s> Main Program }
PROCEDURE PROSES;
BEGIN

assign(ff,'data.dum');
rewrite(ff);

DATADUMP(FF);
FOR I := 1 TO NUMPER DO LOADSYS[I] :=
PERIOD_DATA[I].SYSLOAD;
IF CENTANG THEN BEGIN
FOR J := 1 TO NUMUNT DO BEGIN
UNTCOST[J] := (UNTEC3[J] + UNTEC2[J]*UNTMAX[J]*(1.0-
UNTSR[J]/100)
+
UNTMAX[J]*UNTMAX[J]*UNTEC1[J]*(1.0-
UNTSR[J]/100)*(1.0-UNTSR[J]/100))*UNTFUELCOST[J];

UNTEM[J] := (UNTEM3[J] + UNTEM2[J]*UNTMAX[J]*(1.0-
UNTSR[J]/100)

```

+ UNTMAX[J]\*UNTMAX[J]\*UNTEM1[J]\*(1.0-  
UNTSR[J]/100)\*(1.0-UNTSR[J]/100));

if cweight=3 then PRCPEN[J] :=UNTEM[J] / UNTCOST[J];  
if cweight=2 then PRCPEN[J] := UNTCOST[J] / UNTEM[J];  
if cweight=1 then PRCPEN[J] :=1;  
if cweight=0 then PRCPEN[J] :=1;  
END;

END

ELSE BEGIN

FOR J := 1 TO NUMUNT DO BEGIN

UNTCOST[J] := (UNTEC3[J] + UNTEC2[J]\*UNTMAX[J]\*(1.0-  
(SPR/100))  
+ UNTMAX[J]\*UNTMAX[J]\*UNTEC1[J]\*(1.0-(SPR/100))\*(1.0-  
(SPR/100)))\*UNTFUELCOST[J];

UNTEM[J] := (UNTEM3[J] + UNTEM2[J]\*UNTMAX[J]\*(1.0-  
(SPR/100))  
+ UNTMAX[J]\*UNTMAX[J]\*UNTEM1[J]\*(1.0-(SPR/100))\*(1.0-  
(SPR/100)));

if cweight=3 then PRCPEN[J] :=UNTEM[J] / UNTCOST[J];  
if cweight=2 then PRCPEN[J] := UNTCOST[J] / UNTEM[J];  
if cweight=1 then PRCPEN[J] :=1;  
if cweight=0 then PRCPEN[J] :=1;  
END;

END;

A:=PRCPEN;

SORT(A);

IF CENTANG THEN

FOR I := 1 TO NUMUNT DO UNTMAX[I] := UNTMAX[I]\*(1-  
UNTSR[J]/100)

ELSE

FOR I := 1 TO NUMUNT DO UNTMAX[I] := UNTMAX[I]\*(1-  
SPR/100);

FOR I := 1 TO NUMUNT DO

FOR J := 1 TO NUMUNT DO BEGIN

IF A[I] = PRCPEN[J] THEN BEGIN

NUNTMAX[I] := NUNTMAX[I-1] + UNTMAX [J];

END;

```

END;

STATE := [];
FOR I:=1 TO NUMUNT DO STATE := STATE + [I];
I := 0;
WHILE (STATE <> []) DO BEGIN
    I := I+1;
    UNTSTAT[ I ] := STATE;
    BORROW := TRUE;
    J := NUMUNT;
    WHILE BORROW DO
        IF (J IN STATE) THEN BEGIN
            STATE := STATE - [J];
            BORROW := FALSE
        END
    ELSE BEGIN
        STATE := STATE + [J];
        J := J - 1
    END;
    NUMSTAT := I;
END;
I:=1;
WHILE (STATE <> UNTSTAT[I]) AND (I < NUMSTAT) DO
I:=I+1;
IF (STATE = UNTSTAT[I]) THEN
    INITSTAT:=I
ELSE BEGIN
    NUMSTAT := NUMSTAT + 1;
    INITSTAT := NUMSTAT;
    WRITELN(FF, 'INITIAL STATE NOT IN PRIORITY LIST,
ADDING STATE TO LIST');
    UNTSTAT[ INITSTAT ] := STATE;
END;
IF (NMAX > NUMSTAT) THEN NMAX := NUMSTAT;
IF (XMAX > NUMSTAT) THEN XMAX := NUMSTAT;
WITH PERIOD_DATA[0] DO BEGIN
    FOR I := 1 TO NUMSTAT DO BEGIN
        FCOST[ I ] := INFINITY;
        FOR J := 1 TO NUMUNT DO UNTSTATNOW[ I, J ] := 0;
    END; { I }

```



```


FCOST[ INITSTAT ] := 0.0;
PCOST[ INITSTAT ] := 0.0;
PATH[ INITSTAT ] := 0;
UNTSTATNOW[ INITSTAT ] := UNTSTATINI;
END;
CONTINUE := TRUE;
K := 1;
WHILE (K <= NUMPER) AND CONTINUE DO WITH
PERIOD_DATA[K] DO BEGIN
    UNTSTATLAST := UNTSTATNOW;
    FOR I := 1 TO NUMSTAT DO BEGIN
        FCOST[ I ] := INFINITY;
        EMCOST[ I ] := INFINITY;
        PATH[ I ] := 0;
        FOR J := 1 TO NUMUNT DO UNTSTATNOW[ I, J ] := 0;
    END; { I }
    IF (K = 1) THEN BEGIN
        NLIM := 1;
        SETL[ 1 ] := INITSTAT;
    END
    ELSE
        BESTSTATES(SETL,PERIOD_DATA[K-1]
        ,FCOST,NLIM,NMAX);
    FOR J := 1 TO NUMUNT DO BEGIN
        IF LOADSYS[K]<NUNTMAX[J] THEN BEGIN
            if cweight = 3 then PR:=A[J];
            if cweight = 2 then PR:=A[J];
            if cweight =1 then PR:=1;
            if cweight =0 then PR:=1;
        BREAK;
        END;
    END;
    WRITELN(FF,' Price Penalty Factor = ',PR:4:4);
    EDC;
    writeln(ff,' ');
    BESTSTATES(SETX,PCOST,XLIM,XMAX);
    CONTINUE := FALSE;
    FOR X := 1 TO XLIM DO BEGIN
        XSTATE := SETX[ X ];
        FOR N := 1 TO NLIM DO BEGIN
            LSTATE := SETL[ N ];

```

```

FEAS;
IF (VALID = 1) THEN BEGIN
    PATHCOST := PERIOD_DATA[ K-1].FCOST[ LSTATE ] +
SCOST +
        PCOST[ XSTATE ];
    PATHEM := PERIOD_DATA[ K-1].EMCOST[ LSTATE ]+
        ECOST[ XSTATE ];
    AVEPATH := (PATHCOST/PATHEM);
    AVECOST := (PCOST[ XSTATE ]/ECOST[ XSTATE ]);
    IF Cweight=1 THEN BEGIN
        IF (PATHEM <= EMCOST[ XSTATE ]) THEN BEGIN
            FCOST[ XSTATE ] := PATHCOST;
            EMCOST[ XSTATE ] := PATHEM;
            UNTSTATNOW[ XSTATE ] := UNTSTATTEMP;
            PATH[ XSTATE ] := LSTATE;
            CONTINUE := TRUE;
        END;
    END;
    IF CWEIGHT = 0 THEN BEGIN
        IF (PATHCOST <= FCOST[ XSTATE ]) THEN BEGIN
            FCOST[ XSTATE ] := PATHCOST;
            EMCOST[ XSTATE ] := PATHEM;
            UNTSTATNOW[ XSTATE ] := UNTSTATTEMP;
            PATH[ XSTATE ] := LSTATE;
            CONTINUE := TRUE;
        END;
    END;
    IF CWEIGHT = 2 THEN BEGIN
        IF AVEPATH<=AVECOST THEN BEGIN
            FCOST[ XSTATE ] := PATHCOST;
            EMCOST[ XSTATE ] := PATHEM;
            UNTSTATNOW[ XSTATE ] := UNTSTATTEMP;
            PATH[ XSTATE ] := LSTATE;
            CONTINUE := TRUE;
        END;
    END;
    END; { N }
    END; { X }
    K := K + 1;
    END; { K }

```



```
IF CONTINUE THEN FINAL_OUTPUT(ff) ELSE  
  WRITELN(ff,'THERE IS NO SOLUTION');  
  ENDPRINT(ff);  
  close(ff);  
  {-----}  
END;
```

end.